UNIVERSITE BRETAGNE LOIRE / MATHSTIC

université angers

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ D'ANGERS
COMUE UNIVERSITÉ BRETAGNE LOIRE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Traitement des images et du signal.*

Par

« Salma Samiei »

« Contributions to low-cost imaging and machine learning for plant phenotyping. »

Thèse présentée et soutenue à « INRAE - Angers, Bâtiment A », le « 01/10/2020 »
Unité de recherche : LARIS
Thèse N° : 190338

Rapporteurs avant soutenance :
DR. Frédéric Baret INRAE, Avignon, France.
PR. Aymeric Histace ENSEA, ETIS Lab, Cergy-Pontoise, France.

Composition du Jury :
Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du Jury ne comprend que les membres présents

Presidante :	PR. Diana Mateus	LS2N-CNRS, Ecole Centrale Nantes, France.
Examinateur :	PR. Gerhard Buck-Sorlin	ACO - INRAE, Angers, France.
Examinateur :	DR. Astrid Junker	IPK Gatersleben, Germany.
Dir. de thèse :	PR. David Rousseau	LARIS- INRAE, Université d'Angers, France.

Invité(s) :

Co-encadrant	MCF HDR Paul Richard	LARIS, Polytech Angers, France.
Co-encadrant	MCF Etienne Belin	LARIS-INRAE, Université d'Angers , France.

Acknowledgement

I would like to express my special thanks and sincere gratitude to prof. David Rousseau for the unwavering support and belief in me through my Ph.D. and research. The person who brought reality and possibility for all written in the current thesis. His patience, motivation, vision, enthusiasm inspired me, and his valuable advice and guidance helped me all through this journey. He has taught me how to carry out the research and present it as clearly as possible.

I would like to extend my deepest appreciation to my committee members Dr. Frédéric Baret, prof. Aymeric Histace, prof. Gerhard Buck-Sorlin, prof. Diana Mateus, and DR. Astrid Junker, who gently accepted to evaluate my work and attend in my Ph.D. defense.

I am also grateful to associate prof. Pejman Rasti, my love, and the person whose help cannot be underestimated. His constructive criticism, insightful suggestions, and unparalleled support and guidance made me succeed in this quest.

I also had a great pleasure of working with François Chapeau-Blondeau, Etienne Belin, Paul Richard, Natalia Sapoukhina, Julia Buitink, Gilles Galopin, Hervé Daniel who extended a great amount of assistance and support.

Thanks should go to all my friends and colleagues in the ImHorPhen team, INRAe, and Polytech Angers.

I am forever indebted to my parents for giving me the opportunities, support, love and experiences that have made me who I am. I would like to express my gratitude to my beloved sister (and my coming soon niece), my brothers, my adorable nephew, and my in-laws for their support and encouragement, which never let me down. I heartfully appreciate my love, the two joys of our life, Dani and Tedi, and also the new member of our family, (my unborn baby), who we believe will bring all the blessings to our life. Well, finally I dedicate all my works and research to my beloved family.

I would like to express my special thanks and sincere gratitude to prof. David Rousseau for the unwavering support and belief in me through my Ph.D. and research. The person who brought reality and possibility for all written in the current thesis. His patience, motivation, vision, enthusiasm inspired me, and his valuable advice and guidance helped me all through this journey. He has taught me how to carry out the research and present

it as clearly as possible.

I would like to extend my deepest appreciation to the jury and my committee members Dr. Frédéric Baret, prof. Aymeric Histace, prof. Gerhard Buck-Sorlin, prof. Diana Mateus, and DR. Astrid Junker, who gently accepted to evaluate my work and attend in my Ph.D. defense.

I also had a great pleasure of working with François Chapeau-Blondeau, Etienne Belin, Paul Richard, Pejman Rasti, Natalia Sapoukhina, Julia Buitink, Gilles Galopin, Hervé Daniel who extended a great amount of assistance and support.

Thanks should go to all my friends and colleagues in the ImHorPhen team, INRAe, and Polytech Angers.

I am forever indebted to my parents for giving me the opportunities, support, love and experiences that have made me who I am. I heartfully appreciate my love and finally I dedicate all my works and research to my beloved family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

To introduce our scientific contributions to the interdisciplinary topic of plant phenotyping with imaging and machine learning, we start this thesis by defining what computer vision-based plant phenotyping is and why we care about it. We continue by discussing our motivation to focus on reducing the cost of imaging and machine learning by analyzing what is costly in each element of the imaging chain and the current bottleneck to be addressed. We conclude this short introduction chapter by explaining the organization of the thesis.

1.1 Computer vision-based plant phenotyping

Plant phenotyping corresponds to the measurement of any meaningful observable resulting from the interaction of environment and genotype, as illustrated in Fig 1.1. The environment may include the natural income of nutrients for the plant (light, water, atmosphere, soil). The environment may also include the impact of surrounding plants and microorganisms with biotic interactions (adversarial and mutualistic) or plant-plant interactions (in an agro-ecological multi-species strategy). Plant phenotyping can be done in controlled environments or in more challenging natural conditions. This topic has attracted much attention since the mid-2000s because of the massive increase in throughput of the genomic tools. In order to perform genotype-environment analysis at similar throughput, the bottleneck in plant science was pointed to be phenotyping. Also, plant phenotyping is useful for users to monitor their installation and have precise information on the expected yield.

Plant phenotyping, when performed manually, is extremely costly in terms of manpower. Also, manual assessment, as any tiring and repetitive task, is prone to subjectivity and can lead to uncertainty on the measurement. In addition, phenotyping based on human vision inspection limits the observations to the visible spectrum range. This situation met in the mid-2000s the democratization of the access of imaging systems available for

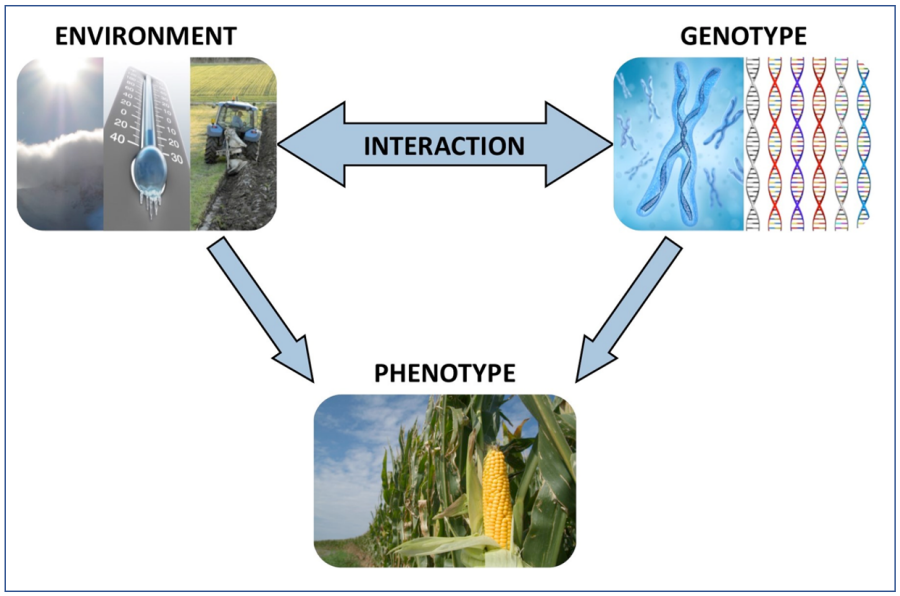


Figure 1.1 – The interaction between genotype and environment conjointly determine plant phenotype expressions. Image copyright International Plant Phenotyping Network (IPPN).

proxy-detection.

At an international level, the initiative of a network of phenotyping centers (FPPN in France, EPPN in Europe, IPPN at a worldwide scale) was launched. The first round of investigations consisted of identifying the best imaging systems adapted to plant imaging. Some centers somehow developed an approach mimicking biomedical imaging with highly accurate imaging systems and low-throughput of individual plant carried, for instance, under X-ray or PET-MRI systems. Alternatively, some centers investigated plant imaging especially useful for plant phenotyping while compatible with higher throughput. Such imaging systems mostly include RGB, LiDAR, thermal, hyperspectral, and fluorescence imaging. Despite the diversity of traits to be measured on plants, the type of plants, their stage of development, or the imaging system used, all computer vision-based plant phenotyping methods share a common framework, as shown in Fig. 1.2.

Light is sent onto the scene, including the plant, to be phenotyped. The interaction of the light with the scene is captured by an optic and sent onto an imaging sensor. Image processing is performed to extract the targeted phenotypic information from the scene. If the types of phenotypic traits to be measured are well-defined from the acquisition step, it is possible to optimize the computer vision system's cost, as indicated by the feedback

arrow in Fig. 1.2. It is the general approach followed in this thesis.

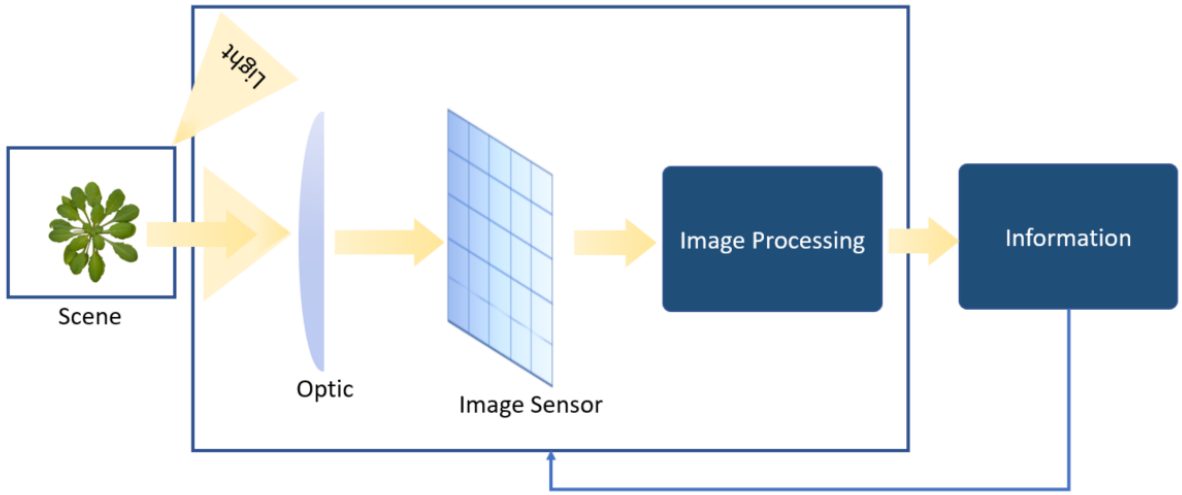


Figure 1.2 – Plant imaging framework. The feedback arrow indicates that the prior knowledge of the targeted informational task can be used to optimize each step of the framework to reduce the global cost of phenotyping.

Several reasons motivate optimization of computer vision-based plant phenotyping in order to reduce its cost. First, for simplicity reasons inspired by the Occam razor principle, a physical system should not be oversized. Second, lowering the cost of the system will enable a better translation of the phenotyping systems to farmers for wide dissemination. Third, the lower-cost system will be easier to replicate on different phenotyping platforms or even on a single platform for parallelization. This reason is especially important for the assessment of plant phenotyping in controlled environments, as considered in this thesis and illustrated in Fig. 1.3.

Indeed, different scenarios may be applied to plant imaging in controlled conditions as illustrated in Fig. 1.3 either sensing equipment moves towards the samples (sensor-to-plant), or the plants are transported towards the cameras (plant-to-sensor), or an alternative approach consists of using a grid of sensors. The plant-to-sensor scenario (middle in Fig. 1.3) allows optimized image acquisition conditions in dedicated cabinets with top-and-side views, with high-resolution sensors and specific illumination conditions. However, this scenario demands to move the plants. This may be invasive specially when studying biotic or abiotic stress because movement itself can stresses plant and can result in cross contaminations. Moreover, this scenario does not allow synchronize acquisitions on large population of plants. The sensor-to-plant scenario (left in Fig. 1.3) keeps the

plants at their place, and top-view imaging equipment screens growing areas. A robotic arm may include the flexibility to acquire images from different point-of-views. However, similarly to the plant-to-sensor scenario, the sensor-to plant scenario is not a synchronized and may create shadows on the plant. At last, the grid-of-sensors (right in Fig. 1.3) scenario solves the problems of speed, synchronization and it remains non invasive. The drawback of this grid-of-sensors scenario may be the cost of the replication of the sensors, unless low-cost sensors are used.

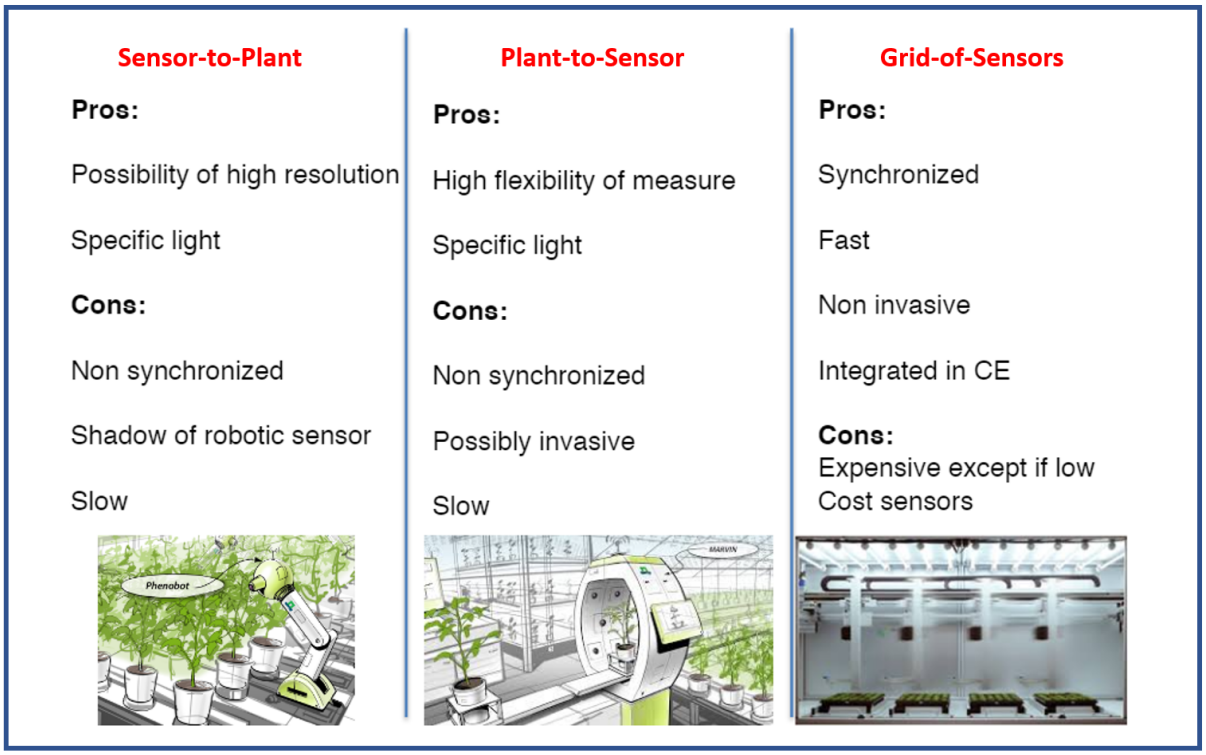


Figure 1.3 – Pros and cons of the different plant imaging scenarios in a controlled environment. Image copyright PhenoKey.

1.2 Toward low-cost computer vision-based plant phenotyping

The cost of a computer vision-based phenotyping can be impacted by each step of the framework of Fig. 1.2. We shortly discuss and illustrate possible ways to optimize each of these steps in this section.

The first step in the plant imaging system comes from selecting the scene's observation scale. Some especially interesting choices are the ones which enables the acquisition of multiple plants in a single image and the extraction of the phenotyping information from a single snapshot. A well-suited imaging geometry for this consists of observing plants from the top view, as illustrated in Fig. 1.4 with various observation scales. This will be the imaging geometry chosen in the thesis. Producing and growing plants cost money and time. As a consequence, working on small plants (seedling), phenotyped at their early stages of life, corresponds to a situation where low-cost plant phenotyping is the most likely to be efficient when including the cost of plant production. In this thesis, we will mainly focus on small plants observable from a single view (side or top).

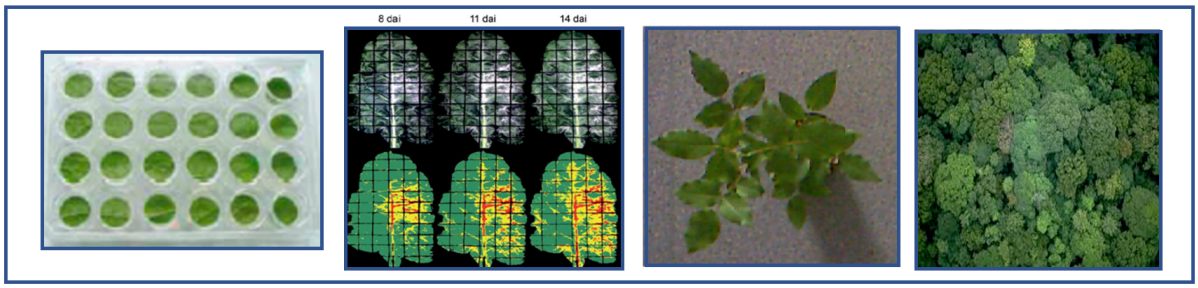


Figure 1.4 – Various observation scales from the top view. From left to right, respectively: foliar disk in vitro, large single leaf held by a metallic grid, short single plant to flat leaves, canopy.

A second step in the plant imaging comes with the choice of light. While plants have their own needs in terms of light, computer vision requires contrast in acquired images, which can be optimized with the selection of the most appropriated wavelengths that are sent onto the scene. A range of available technological solutions is presented in Fig. 1.5. Lower cost approach comes when sufficient contrast is accessible with standard lighting conditions for plants and standard cameras. This will be the situation considered in the plant phenotyping use-cases addressed in this thesis.

Optics is another step that needs to be considered in plant imaging. Distortion-free optics are costly. There is often a tradeoff between the field of view and the amount of distortion since a larger field of view comes with more distortions. It is however, possible to compensate for these distortions when some objects with prior knowledge of shape are located in the scene. This enables to reduce the cost of optics without actual impact on the quality of the extracted information. This will be the situation considered in this thesis.

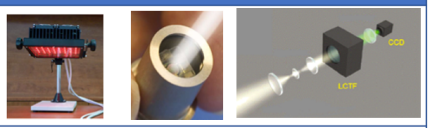
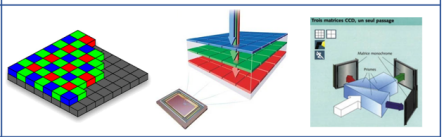
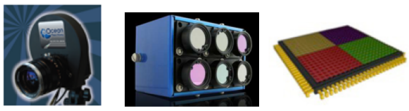
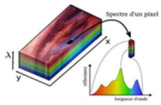
Type of sensors	Number of channels	Value at each pixel	Filters	Imaging systems
Black/White (monochrome)	1 channel	Gray level	Open choice (visible/NIR)	
Color	3 channels	RGB level	Imposed (RGB) or on-demand	
Multispectral	Many channels (4-20 bands)	Spatial and spectral	Open choice (visible/NIR)	
Hyperspectral	Many channels (+20 bands)	Spatial and spectral	-	

Figure 1.5 – Technological approaches to optimize contrast by wavelength selection in plant imaging.

The next step in plant imaging is the choice of the sensor. For a given same question, a variety of imaging technologies can be deployed. The chosen sensor should take into account contrast, but also the requested resolution and dynamic. Figure 1.6 and Table 1.1 extracted from [1] illustrate such a choice for a task of azimuth leaf orientation measurement. The laser scanner, while providing a leaf segmentation of higher accuracy, does not improve the quality of the leaf orientation compared to the much lower cost Kinect sensor.

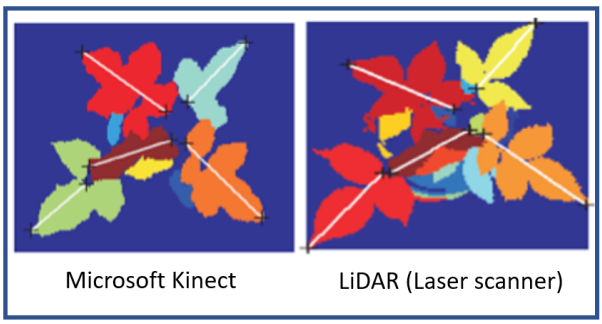


Figure 1.6 – Azimuth leaf angle measured with different sensors for the same task of leaf orientation determination. Reproduced from [1].

The development of the image processing step is currently identified as the bottleneck

Depth Camera	Kinect	LiDAR
Azimuth 1	46.43°	48.04°
Azimuth 2	125.16°	120.65°
Azimuth 3	19.94°	16.36°
Azimuth 4	154.52°	160.73°

Table 1.1 – Similar result with both sensors for azimuth leaf orientation measurement.

in image based plant phenotyping. It has to be developed in order to extract the targeted information. One can define three types of informational tasks. A first type is when the phenotypic trait to be measured is exactly defined. The plant phenotyping questions which are located in this category can be solved easily by classical handcrafted image processing approaches. The cost of such informational tasks can be the lowest since we know what kind of information we are seeking and this can serve to optimize each element of the plant imaging framework. The second type is when the feature space on which to perform the task is not known but we have prior knowledge of phenotypic difference. This category leads to implement supervised machine learning algorithms. The last type is when no prior knowledge of phenotyping difference can be assumed. In this case, unsupervised machine learning approach can be a chosen approach in order to identify possible clusters in the observed populations of plants. In this thesis, we will mostly focus on the second type, the situation which can be solved with supervised learning and mainly discuss how to reduce the cost of this machine learning approach for plant imaging.

Supervised learning is the machine learning approach that maps input data to an output based on example input-output pairs. Supervised learning requires annotated dataset to infer a learning algorithm. This ground-truth dataset is used to predict the output for other unlabeled dataset through the use of machine learning algorithms. Modern supervised machine learning applied to image processing is a revolution. In early 2000, developing an image processing application for plant phenotyping tasks would require almost one year of research to define the prototype and implementation and deployment of the solution. Since 2012, by the breakthrough in supervised deep learning, the required time to apply research and development to implement a plant phenotyping solution fell off to couple of days. However, supervised deep learning, has some hidden costs with the necessity to provide large amount of ground-truth data to train the model and the huge computational cost for the training of the model. In this thesis, we will discuss, via specific use cases, ways to reduce these costs of supervised deep learning.

1.3 Contributions and organisation of the document

Recently in [2], cost-efficient plant phenotyping was investigated in detail for field experiments, but not in controlled environment. As also underlined in [3], the study identified the current bottleneck of plant phenotyping as the development of image processing. In this thesis, we revisit low-cost phenotyping and push forward its analysis by focusing on imaging in a controlled environment and producing contributions on how to reduce the cost of image processing itself. As motivated in the previous section, we mainly focus our effort on supervised machine learning approaches applied to use cases with low-cost imaging systems gazing at plants with a single view (top or side view).

The document is organized as follows. In Chapter 2, we will first focus on reducing the cost of plant imaging at the imaging system level. To this purpose, we present the single-boards computers we used to design networks of sensors. We review the current state-of-the-art technologies for these mini-computers and present their applications in the literature of plant phenotyping. Similarly, we then present low-cost cameras operating in the visible or infrared spectrum to perform reflectance imaging or range imaging at a low cost. After this state of art, we discuss the specific interest of seedling as a stage of plant development especially suited to benefit from these low-cost imaging systems. We present our contributions to this stage of development operating on individual seedling or on a group of seedling observed as a surface called the canopy. In chapter 3, we then focus on reducing the cost of machine learning. We identify the main bottleneck as the time of annotation and the computational cost. After a state-of-the-art of the current approaches, we present our contribution to speed up annotations with the help of eye-tracking technologies and to perform deep learning at low computational costs. This thesis has been performed on the phenotyping platform of Angers. The biological use cases taken for illustration of the methodological contributions are multiple and represent the wide variability of computer vision problems encountered in plant phenotyping.

Chapter 2

Low-cost imaging

Plant imaging systems can extend phenotyping capability, but they require a platform to handle high-volume data. However, commercial platforms that make consistent image acquisition easy are often cost-prohibitive to many laboratories and institutions. There is also no such thing as a "one-size-fits-all" phenotyping system; different biological questions often require different hardware configurations. Therefore, to make more accessible high-throughput phenotyping methods, low-cost single-board computers (SBCs) and imaging sensors can be used to acquire plant image data[4].

This chapter investigates the available low-cost technologies (imaging sensors and single-board computers) to perform image-based plant phenotyping. First, we review the state-of-the-art technologies and their applications in image-based phenomics. Then, we present our contributions with low-cost imaging devices coupled with single-board computers.

2.1 Low-cost state-of-the-art technologies

This section will describe state-of-the-art plant phenotyping technologies to enable academic and commercial plant scientists to address complex problems in plant and agricultural science. Initially, we will represent the various kinds of low-cost single-board computers. We will continue by explaining low-cost visible-light imaging sensors and 3D imaging systems.

2.1.1 Single-board computers

Single-board computers (SBCs) or mini-computers are small computing devices that can be used for various purposes. A single-board computer encompasses all the elements of a computer, such as memory, input/output, and a microprocessor embedded in a single circuit board. In contrast to conventional computers, single-board computers are indepen-

dent of expansions for functionality and are self-contained. Because of this feature, they are frequently used in rack systems, which allows for reliable and fast integration into a system and relatively easy to swap one out for an other if a computer needs to be replaced. They are lightweight and compact, which allows them to be embedded in places where space is minimal. Mini-computers are very efficient and low power consumption, which makes them very cost-effective solutions for research purposes likewise, industries. There has been much interest recently in designing image-based phenomics platforms powered by this technology and its relevance sensors, which is discussed in the following.

Single-board computers can be divided into two main categories, open-source and proprietary. Open source SBCs give access to both hardware design and layout plus the source code used on the board, and this is ideal for understanding how the software and hardware operate and adapt them to end-design requirements. Proprietary SBCs, on the contrary, are generally designed for use in end applications or as a reference to be evaluated.

SBCs can be used for several purposes, including personal, research, and educational purposes, as well as rapid prototyping development in the Internet of Things (IoT) or, automates human tasks with high performance. Choosing the right SBC for an application requires many considerations. We provide an overview of the available SBCs in the market by considering some criteria in Table 2.1. Nevertheless, there are other options, such as power, backward pin compatibility, storage, and more, which must be considered to choose SBC according to the necessity.

Current SBCs come with a wide diversity of processor types, most with GPUs on-board and the brand-new generation contains an Edge TPU or NPU coprocessor which is ideal for prototyping the projects that demand fast on-device inferencing for machine learning models (NVIDIA Jetson Nano, Coral Dev Board, Rock Pi N10). The most prevalent form of operating system used on SBCs is Linux-based. The cost might vary in the range of 10€ to 300€, depending on all the options mentioned heretofore.

SBCs	CPU	GPU	Memory	OS	Connectors	Network (built in)	Price
Raspberry Pi Zero W	Broadcom BCM2835 (1x ARM1176JZF @1GHz)	VideoCore IV dual-core	512MB	Raspbian (Linux-based)	Mini HDMI, 2 x USB OTG	Wi-Fi, Bluetooth	~ 10€
Onion Omega 2Plus	580 MHz MIPS	n/a	128 MB	Linux	1 x USB 2.0	Wi-Fi	~ 13€
Rock64 Media Board	Rockchip RK3328 (4x Cortex-A53 @ 1.5GHz)	Mali-450 MP2	up to 4GB DDR3	Android , Linux	2 x USB 2.0 1x USB 3.0	1Gb Ethernet	~ 44€
PocketBeagle USB key-fob computer	Octavo Systems OSD335x SiP (1x Cortex-A8 @ 1GHz)	PowerVR SGX530	512MB	Linux	1 x USB OTG	-	~ 25€
Pine A64-LTS	Allwinner R18 (4x Cortex-A53 cores @ 1.2GHz)	Mali-400 MP2	2GB DDR3	Android , Linux	2 x USB 2.0	1Gb Ethernet, Wi-Fi, Bluetooth	~ 44€
Banana Pi M64	Allwinner A64 (4x Cortex-A53 @ 1.2GHz)	Mali-400 MP2	2GB DDR3	Android , Linux	2 x USB 2.0 HDMI	1Gb Ethernet Wi-Fi, Bluetooth	~ 63€
Odroid-C2	Amlogic S905 (4x Cortex-53 @ up to 1.5GHz)	Mali-450 MP2	2GB DDR3	Android , Linux	4 x USB 2.0	1Gb Ethernet	~ 46€
Orange Pi Plus2	Allwinner H3 (4x Cortex-A7 @ 1.6GHz)	Mali-400 MP2	2GB DDR3	Android , Linux	4 x USB 2.0 HDMI	1Gb Ethernet, Wi-Fi, Bluetooth	~ 21€
Rock Pi 4 Model B	Rockchip RK3399	Mali T860MP4	1- 4GB DDR4	Android , Linux	2 x USB 2.0 1 x USB 3.0 HDMI	1Gb Ethernet Wi-Fi, Bluetooth	~ 19€
NanoPC-T3 Plus	Samsung S5P6818 (8x Cortex-A53@ 1.4GHz)	Mali-400 MP	2GB DDR3	Android , Linux	4 x USB 2.0 HDMI	1Gb Ethernet, Wi-Fi, Bluetooth	~ 99€
Raspberry Pi 3B	Broadcom BCM2837 (4x cortex-A53 @ 1.2GHz)	VideoCore IV @ 400 Mhz	1GB DDR2	Raspbian (Linux-based)	4 x USB 2.0 HDMI	100Mb Ethernet, Wi-Fi, Bluetooth	~ 35€
Raspberry Pi 4B	Broadcom BCM2711 (Cortex-A72 @ 1.5 GHz)	VideoCore VI	4GB DDR4	Raspbian (Linux-based)	2 x USB 3.0 2 x USB 2.0 2x Micro HDMI	1Gb Ethernet, Wi-Fi, Bluetooth	~ 55€
Odroid-XU4	Samsung Exynos5422 (4x Cortex-A15 @ 2.0GHz and 4x Cortex-A7 @ 1.6GHz)	ARM Mali-T628 MP6	2GB DDR3	Android , Linux	1 x USB 2.0 2 x USB 3.0 HDMI	1Gb Ethernet	~ 59€
Asus Tinker Board S	Rockchip RK3288 (4x Cortex-A17 @ 1.8GHz)	ARM Mali-T760	2GB DDR3	Android , Linux	4 x USB 2.0 HDMI	1Gb Ethernet, Wi-Fi, Bluetooth	~ 55€
LattePanda	Intel Cherry Trail Z8350 Quad Core 1.8GHz	Intel HD Graphics @200-500 Mhz	2GB DDR3	Windows 10	2 x USB 2.0 1 x USB 3.0 HDMI	100 Mb Ethernet, Wi-Fi, Bluetooth	~ 109€
NVIDIA Jetson Nano Best flexibility for ML	Quad core A57 @ 1.43 GHz	128-core NVIDIA Maxwell™	4GB DDR4	JetPack (Linux-based)	4 x USB 3.0 2 x HDMI	1Gb Ethernet	~ 89€
Coral Dev Board Best for ML with TensorFlow	NXP i.MX 8M SOC (quad Cortex-A53, Cortex-M4F)	GC7000 Lite Graphics TPU: Google Edge	1GB DDR4	Mendel (Google)	1 x USB 3.0 HDMI	1Gb Ethernet, Wi-Fi Bluetooth	~ 150€
Minnowboard Turbot Dual Ethernet	Intel Atom E3845 Series @ 4 x 1.91 GHz	Gen 7 (4 Execution Units)	2GB DDR3	Windows, Android , Linux	1x USB 2.0 1x USB 3.0 HDMI	2x 1Gb Ethernet	~ 196€
Rock Pi N10 Best for ML	Dual Cortex-A72 @1.8GHz with quad Cortex-A53@ 1.4GHz	Mali T860MP4 GPU NPU: up to 3.0TOPs computing power	up to 8GB DDR3	Android , Linux	2x USB 2.0 1x USB 3.0 HDMI	1Gb Ethernet	~ 169€
BeagleBoard-X15	Dual-core Sitara AM5728 ARM Cortex A15 @ 1.5Ghz	Dual Core SGX544 , 532 MHz	2GB DDR3	Android , Linux	3 x USB 3.0 eSATA HDMI	2 x 1Gb Ethernet	~ 145€
Huawei HiKey 960	Kirin 960 (4 x 2.3GHz ARM A73 cores, and 4 x 1.8GHz ARM A53 cores)	ARM Mali G71 MP8	3GB DDR4	ASOP (Linux-based)	1 x USB 2.0 2 x USB 3.0 HDMI	Wi-Fi , Bluetooth	~ 249€

Table 2.1 – Comparison of single-board computers(SBCs) capacity in terms of the processor (CPU), GPU, memory (RAM), supported operating systems, connection ports, network support, and the price market at the time. Red color shows the mini-computer model used in this work.

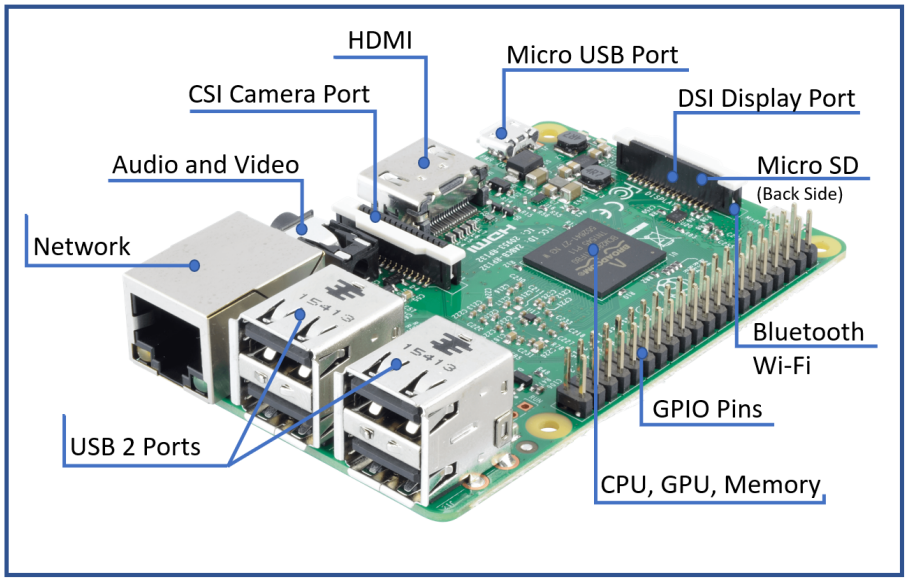


Figure 2.1 – Raspberry Pi version 3B - mini-computer used in seedling growth monitoring in our work on individual observation scale.

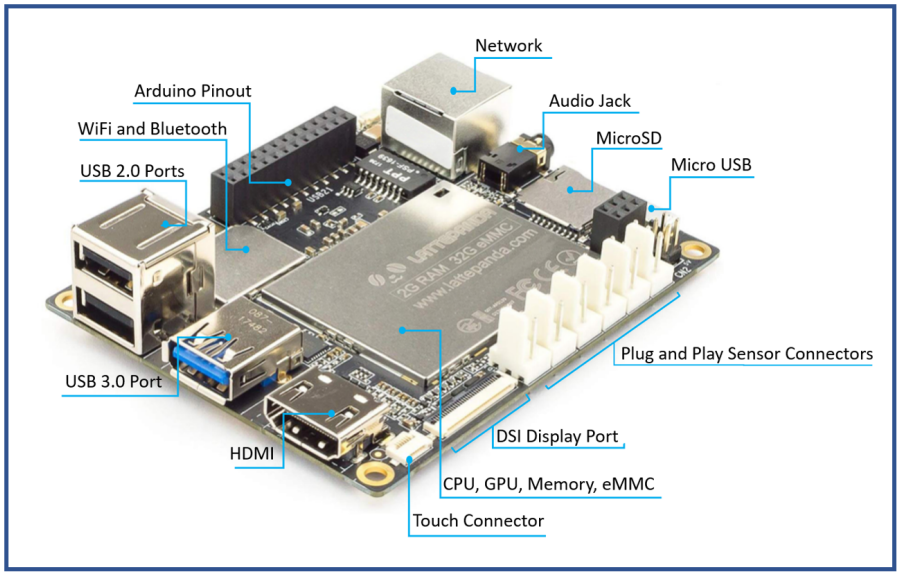


Figure 2.2 – LattePanda - mini-computer used in seedling growth monitoring in our work on canopy observation scale.

Figures 2.1 and 2.2 illustrate the type of SBCs used as low-cost imaging systems in this thesis (section 2.3). In addition to computing devices to develop a low-cost plant phenotyping imaging system, we should consider the type of sensors used. In the next

section, we will explain different available low-cost imaging sensors in more detail and identify the one selected in this Ph.D. document.

2.1.2 Low-cost imaging sensors

A variety of imaging technologies are being used to deploy an automated image acquisition platform to collect data for quantitative studies of complex traits. Substantial low-cost imaging devices are available to measure a phenotype quantitatively through the interaction between light and plants. These devices can be embedded in connected objects such as smart-phones, tablets, or mini-computers or be fixed on various devices such as UAVs, UGVs, or connected sticks.

A key advance in high-throughput phenotyping platforms is the capability to capture plant traits non-destructively by different imaging modalities [5]. This advance permits time-series measurements that are necessary to follow the progression of growth and stress on individual plants. Time-lapse imaging is a valuable tool for recording plant development and can reveal differences that would not be apparent from endpoint analysis.

In the following, available low-cost visible-light imaging systems and depth sensors are described in more detail. We focus on the description of low-cost visible-light (RGB) and depth (3D) imaging sensors available for collecting data relevant to plant phenotyping in the field or controlled environments.

Visible light imaging

Visible-band imaging systems are the fundamental apparatuses for measuring morphological traits (color, shape, size, and texture) of plants. As technology advances, RGB cameras that are suitable for aerial and ground applications have become affordable. Digital single-lens reflex (DSLR) cameras have played a key role in many phenotyping applications, especially in controlled but also field environments. Standard off-the-shelf cameras use silicon-based sensors that are responsive to visible light wavelengths in the 400–1000 nm range. Although, color cameras are further restricted to the 400–700 nm range visible to humans with the inclusion of an infrared-blocking filter. Mounting Cost-effective RGB PiCamera modules to mini-computers are a practical choice for developing time-lapse low-cost phenotyping systems that work effectively in controlled environments [6]. Table 2.2 summarises different examples of low-cost Visible-band imaging systems.

	Sensor	Sensor Resolution	Lens angle-of-view	Price
PiCam V2	Sony IMX219PQ	3280 × 2464	62.2 ° H, 48.8° V	∼ 30€
PiCam V1	OmniVision OV5647	2592 × 1944	53.50 ° H (+/- 0.13), 41.41° V (+/- 0.11)	∼ 25€
Spy Camera (for Raspberry Pi)	Omnivision OV5647	2592 × 1944	54 x 41	∼ 33€
Zero Spy Camera (for Raspberry Pi Zero)	Omnivision OV5647	2592 × 1944	53,50° H 41,41° V	∼ 16€
Pixy2 CMUcam5	Aptina MT9M114	1296 × 976	60° H, 40° V	∼ 75€
OpenMV Cam	Omnivision OV7725	640 × 480	not mentioned	∼ 60€
Spresense Camera Board	Sony CXD5602PWBCAM1	2608 × 1960	65.7° H, 51.6° V	∼ 32€
Spectacles v2 (Snapchat Smart-glasses)	not mentioned	1642 × 1642	115°	∼ 150€
Adafruit MLX90640 (IR thermal camera)	Adafruit MLX90640	32 × 24	wide: 110° H, 70° V narrow: 55°H, 35° V	∼ 60€
FeatherWing (IR thermal camera)	Panasonic AMG8833 GridEYE	8 × 8	7.7 ° H, 8 ° V	∼ 35€
M5StickV (AICamera)	OmniVision OV7740	640 × 480	not mentioned	∼ 28€
ESP32-CAM (AICamera)	OmniVision OV2640	1600 × 1200	not mentioned	∼ 6€
GoPro HERO	Sony IMX277	2560 × 1920	122.6° H, 94° V	∼ 280€
Fish-eye Camera KAYETON	OmniVision OV2710	1600 × 1200	180 °	∼ 52€

Table 2.2 – Different types of low-cost RGB cameras. Red color indicates the RGB sensor used in this thesis.

3D imaging

2D RGB imaging is not robust to various illumination conditions and occlusion of plant organs. These include overlapping leaves and branches or shadowing when we are observing them in the canopy and at mid-growth stages. 3D imaging provides a proper low-cost solution to meet these challenges. Moreover, 3D plant phenotyping allows researchers to reconstruct plant architecture and measuring complex plant morphologies. For example, leaf surfaces can be reconstructed in 3D, and the depth information on the leaf allows its occlusion or shadowing effects to be evaluated, by assisting with the interpretation of conventional images of the plant [7].

Several techniques can produce 3D plant models. We classify these techniques into two groups, including active imaging systems such as light detection and ranging (LiDAR; or laser scanner) sensors, time-of-flight (ToF) cameras, and structured light projection and passive imaging systems inclusive binocular stereo-vision and multi-view stereo-vision. The passive 3D imaging systems use computational volumetric reconstruction algorithms to recover the 3D model. The 3D imaging system's speed depends on the plant morphological traits and the challenges the imaging system needs to overcome to generate the 3D model [8]. However, not each 3D imaging technique is sufficiently fast to provide a high-throughput system. Some information will be extracted from the models, such as plant height, leaf area, and shapes, which are helpful in plant recognition, stress detection, plant function, and agricultural traits. Therefore 3D imaging systems are a well-suited tools as these devices enable exact geometry and growth measurements.

The most affordable conventional method of acquiring 3D data is stereo-vision. Stereo cameras use the correspondence between images to calculate distances in the form of disparity maps and provide estimates of depth for objects in the image [9]. Stereo analysis has been successfully used in controlled environments, essentially to construct 3D models of individual plants. A multi-view stereo-vision system uses multiple cameras to have different perspectives from the plant. All the images of the object of interest will be merged into one full 3D point cloud image. Although it is not a low-cost solution compared with stereo-vision, it is very accurate to reconstruct the 3D models with high accuracy.

LiDAR (Light Detection And Ranging) is a remote sensing technology that measures the sensor's distance to the plant by illuminating pulsed laser dots to the target and measuring the reflected pulses. LiDAR is a fast image acquisition technology that allows scanning at high frequency. Using this technology, which is available in low-cost plant phenotyping, can be performed in the field at night since it is light-independent technology.

LiDAR devices can be used to acquire multi-source phenotypic data during the entire crop growing period and extract important plant morphological traits, such as plant height, plant width, leaf area, leaf length, leaf width, and leaf inclination angle for plant biological and genomic studies [1]. However, It needs calibration before each acquisition, and it is not accurate for detecting small plant organs or seedling, and unsharpened edges of plants like leaves, for instance.

The structured light sensors measure the pattern and the shifts in the pattern to reconstruct the object's distance in the scene. Often the projectors used for measuring the pattern are working in the near-infrared spectrum. This sensor is susceptible to sunlight, and it is only suitable for monitoring plants at night or in light-controlled environments. The first version of the Microsoft Kinect uses structured light technology to generate depth information from the scene. This low-cost depth sensor can be used in plant phenotyping to project an infrared pattern onto the plant to assess the depth information.

Time-of-flight (ToF) camera is a range imaging camera system that employs time-of-flight techniques to resolve distance between the camera and the object for each point of the image, by measuring the round trip time of an artificial light signal provided by a laser. These sensors are generally low resolution compared to RGB cameras but can offer depth without the computation of stereo camera setups. ToF cameras work well in low-light but are susceptible to noise in direct sunlight the same as the structured light sensors. Despite their relatively low resolution and sensitivity to ambient light, ToF is generally favorable for determining features in outdoor agricultural settings. The technology used in Microsoft Kinect v.2 to compute the depth is based on the ToF sensor. Figure 2.3 illustrates different types of 3D sensors and cameras explained in this section.

We briefly introduced different types of active depth sensors, as imaging systems that shine light onto the scene. The light reflected from the scene is used to build the depth image, whether by measuring the time-of-flight between emission and reception or by measuring the deformation of the spatially structured lighting pattern. Some devices can be associated with a conventional RGB imaging system and the depth sensors to produce after registration, a four components RGB-D image. For instance, in Microsoft Kinect systems [1] capturing process consists of obtaining a colored image (RGB) and performing a depth measurement with a structured light technique in the first version or ToF in the second version. RGB-D cameras give a similar output as stereo cameras but by less computation, and they have limited range due to the necessary matching between

1. `https://developer.microsoft.com/en-us/windows/kinect/`

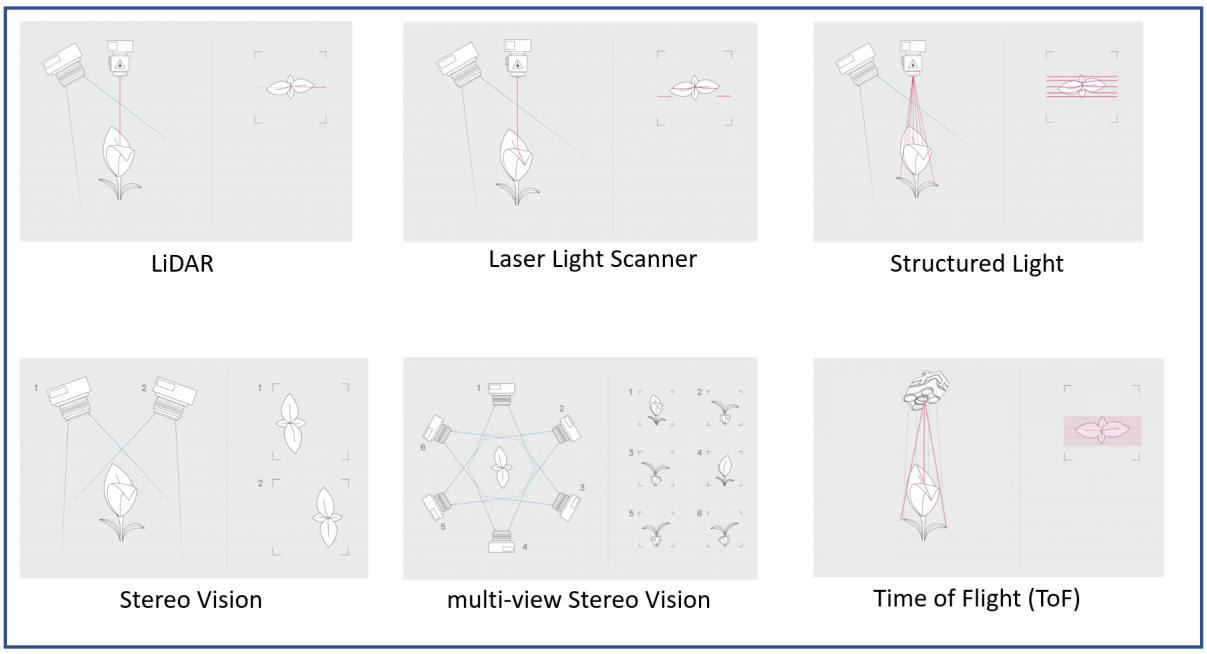


Figure 2.3 – Different types of 3D sensors. Image copyright Phenospex.

depth and RGB cameras. Nevertheless, it is possible to use the output of the depth sensor separately. Figure 2.4 shows the comparison of the depth map generated by the structured light sensor (Kinect v.1) and the ToF sensor (Kinect v.2). Table 2.3 illustrates different low-cost 3D imaging devices and indicates the technology used in this thesis.

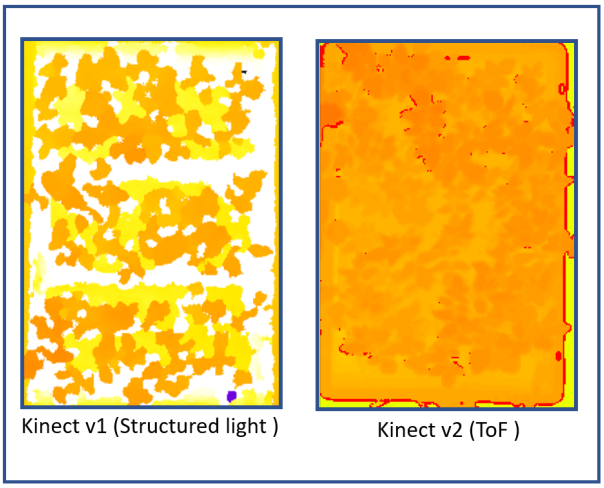


Figure 2.4 – Comparison of the depth map generated by the structured light sensor (Kinect v.1) and the ToF sensor (Kinect v.2).

	Sensor	Depth Range	Infrared(IR) / 3D Resolution	RGB Resolution	Lens field-of-view	Price
DepthEye 3D visual	OPT8320, ToF	0.2∼ 2m	80 x 60 @ up to 1000 fps	N/A	90°D, 75°H, 59° V N/A	140€
LiDAR-Lite v3	Structural light	20 x 48 x 40 mm	+/- 1 cm (.4 in.)	N/A	N/A	130€
Garmin LiDAR-Lite v4	ToF telemetry module	5 cm to 10 meters	N/A	N/A	N/A	66€
IFM O3X100	PMD 3D ToF-Chip	N/A	224 x 172	N/A	60°H, 45°V;	565€
DUO3D	stereo-vision integration into ROS	0.1 - 4 m	224 x 171	N/A	N/A 165° Wide Angle	500€
OV2640	Binocular camera	N/A	N/A	1600 x 1200	78°	12€
SceneScan Pro	stereo-vision	N/A	100 Fps	2.0 MP	16° ∼ 91°	–
MYNT EYE D	Embedded stereo camera, IR module, Integrated depth computation	0.3 ∼ 10 m+ 0.5 ∼ 15 m+	1280 x 720 @ 60 fps	2560 x 720 @ 60 fps	121° D, 105° H, 58° V 70° D, 64° H, 38°V	390€
Intel® RealSense™ Camera D415	Active IR Stereo	0.3 to 10 m	1280 x 720 max @ 30 fps	1920 x 1080 max @ 30 fps	63.4° H, 40.4° V (+/-3°)	205€
Stereolabs® ZED™	Embedded stereo camera	1.5 to 20 m	2208 x 1242 max	2208 x 1242 max	96° H, 54° V	349€
Azure Kinect	ToF IR depth camera, Embedded Stereo Camera, IMU sensor	WFOV: 0.25 - 2.21 m NFOW: 0.5 - 5.46 m	640 x 576px @ 30 fps 512 x 512 px @ 30 fps 1024 x 1024 px @ 15 fps 1 megapixel	3840x2160 @ 30 fps 12 megaPixel	NFOW: 75° H, 65° V WFOV: 120° H, 120° V	400€
Microsoft Kinect V2	ToF, Embedded Stereo Camera	0.5 to 4.5 m	512 x 424 @ 30 fps	1920 x 1080 @ 30 fps	70° H, 60° V (Depth) 84.1° H, 53.8° V (RGB)	130€
Microsoft Kinect V1	Structured light Embedded Stereo Camera	0.8 m to 4 m	320 x 240 @ 30 fps	1920 x 1080 @ 30 fps	57° H, 43° V (Depth) 62° H, 48.6° V (RGB)	80€
ASUS® XtionPro™ Live	Structured light	0.8 to 3.5 m	640 x 488 @ 30 fps	1280 x 1024 @ 30 fps	58° H, 45° V	900€
e-Con Systems Tara Stereo Camera	Embedded Stereo Camera		752 x 480 @ 60 fps	N/A	60° H	149€

Table 2.3 – Different low-cost 3D imaging devices. Red Color indicates depth sensor used in this thesis.

2.2 Applications in plant phenotyping

Low-cost technologies (SBCs and cameras) presented in the previous sections have been applied to affordable image-based phenomics in the following recent literature in both controlled and field environments [4, 6, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. In the following, we briefly review these use-cases of low-cost phenotyping platforms.

Studying plant geometry in fields is significantly essential for plant phenotyping application and plant breeding. This use-case was studied for the first time in 2002, by using a low-cost 3D imaging system in the aerial stereo-vision platform developed in [22]. A mobile stereo-vision system with different sensors and a personal computer was mounted to a remote-controlled helicopter for acquiring site-specific stereo-field scenes illustrated in Fig 2.5. Later in 2005, a 3D crop map based on the stereo-processing of these aerial images of a maize field was generated [23].

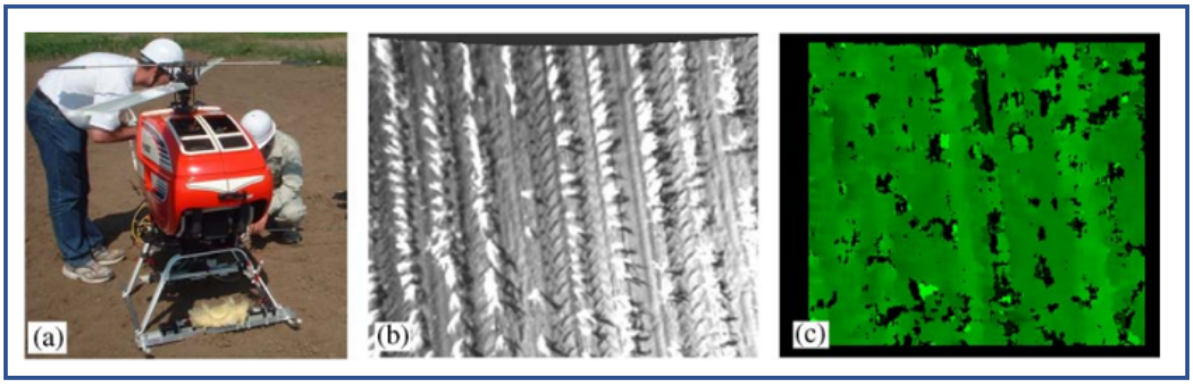


Figure 2.5 – (a) Helicopter-mounted stereo-vision system; (b) acquired crop image; (c) resulting disparity image. Reproduced from [23].

In recent years, due to technical development and the advent of new generations of unmanned vehicles in vision-based plant phenotyping, we can easily access UAVs' aerial images. In [24], a UAV-based imaging platform with using NVIDIA Jetson [2] and different sensors, such as a 3D LiDAR, stereo camera, and GPS antenna used to measure plant height from 3D LiDAR point clouds in real-time. This platform was explicitly focused on imaging row crop environments and analyzing data by machine learning algorithms implemented on SBC. Figure 2.6 illustrates the UAV-based platform with selected sensors.

2. https://developer.nvidia.com/embedded/jetson-tx2-developer-kit

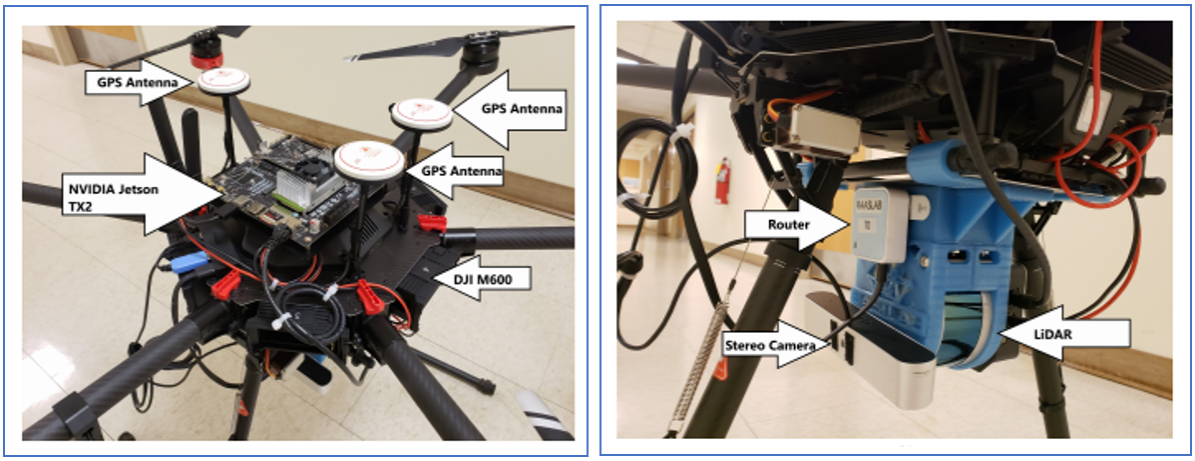


Figure 2.6 – UAV platform with selected sensors connected to the NVIDIA Jetson TX2 SBC on-top and beneath. Reproduced from [24].

In the field, RGB imaging systems are also applicable in different scenarios. For example, the RGB-derived vegetation indexes are presented as the most suitable traits to be measured. For this reason, a set of vegetation indexes' performance was studied by performing the ground (UGV) and aerial(UAV) measurements acquired by the conventional digital camera in [27]. RGB images provide information on the canopy cover and canopy color. For example, the leaf area index (LAI), and light interception can be estimated by color thresholding. Other sophisticated insights can also be extracted by image analysis. For instance, different kinds of biotic and abiotic stresses on plants like water stress or salinity stress could be studied based on the shape, compactness, and solidity of the canopy [28, 29, 30, 31, 32]. Flower density and flowering period of almond trees at the field scale was investigated in [26] by generating colored photogrammetric point clouds using a low-cost (RGB) camera onboard a UAV captured images regularly during the growing period. In another study, PYM (raspberry **P**i p**Y**thon i**M**aging) [17] was designed to measuring the phenotype plant leaf area in a wide diversity of environments in the fields. The method was based on the plant leaf's ability to absorb blue light while reflecting infrared wavelengths.

Phenotyping applications in field environments are a relatively well-covered topic. The useability and flexibility of low-cost and affordable image acquisition systems by using UAVs, UGVs, or connected sticks are studied in a wide range of literature, and different active and passive sensors for agriculture imaging are compared in [25, 33]. In this document, we focus on plant phenotyping in a growth chamber or greenhouse for

experiments under relatively well-controlled conditions. In the following first, some low-cost plant phenotypic platforms designed to address specific phenotyping questions or to acquire images for different purposes in controlled environments will be mentioned. Next, we will cover our contributions to developing an affordable, flexible, and accurate image acquisition system to address two essential plant phenotyping challenges studied in controlled environments such as growth chambers and greenhouses.

In controlled environments, we review the image-based plant acquisition platform designed by low-cost systems and RGB sensors by introducing Phenotiki (Fig. 2.7) developed by [6]. Phenotiki is an affordable top view image-based plant phenotyping system, consisted of a low-cost Raspberry Pi mini-computer attached to the RaspiCam fixed-optics imaging sensor and open-access software. It is an easy-to-use, deploy image-based phenomics platform and freely available to the academic community. Data storage and processing were decoupled from the acquisition. Image data can be transmitted over the local network or the Internet to a centralized repository for analysis. Robust image processing algorithms have been efficiently implemented to enable annotation, detection, tracking, and segmenting plants from the background [34], and counting leaves automatically [35].

The next impressive chamber-based low-cost platform is an automated high-throughput phenotyping pipeline introduced by [10]. It is designed based on affordable imaging systems and image processing algorithms to build 2D mosaicked orthophotos. Off-the-shelf, low-cost digital cameras measure phenotypic traits such as leaf length area and plant vegetation conditions in 2D images. This automated pipeline has cross-platform capabilities and a degree of device independence, making it suitable for various situations. Figure 2.8 illustrates the low-cost imaging platform and orthophoto processing. This platform was used in [11] to quantify 2D and 3D leaf areas for mapping the population of Arabidopsis thaliana and use 2D areas to analyze plant nastic movements and diurnal cycles.

GlyPh [19] is another example of a low-cost RGB platform for high-throughput measurement of plant water-use and growth, to assess the drought tolerance of two soybean genotypes. Top- and side-view images of soybean plants were taken in canopy scale automatically by several digital cameras to measure traits such as height, width, and projected leaf area.

The plant acquisition system can be designed to monitor plants on an individual scale by low-cost sensors. Although the cost of this unique system will be low, it may not be a low-cost system to replicate on different phenotyping platforms or even on a single

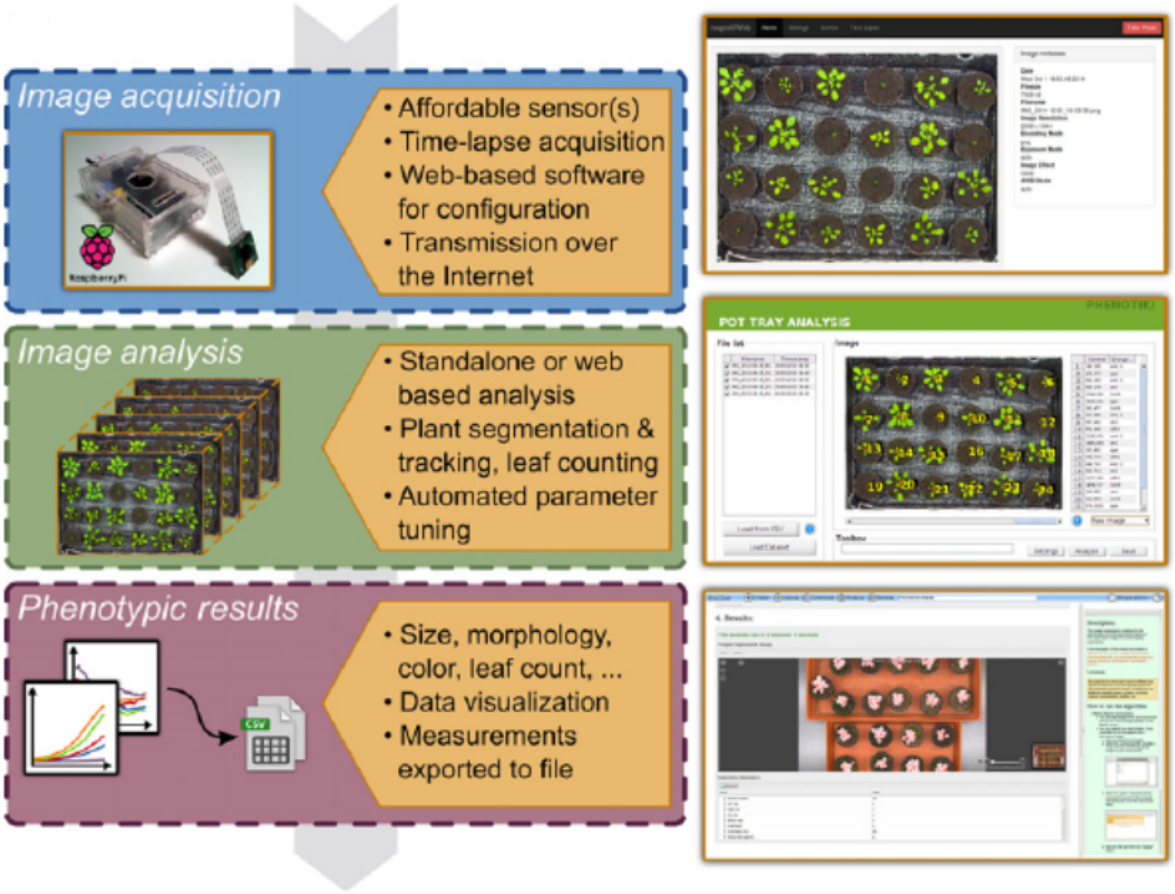


Figure 2.7 – An overview of the Phenotiki system and screen captures showing the graphical user interfaces to operate its hardware and software components. Reproduced from [6].

platform for parallelization. For example, **I**nternet of **L**iving **T**hings (IoLT) [18], is a smart pot platform designed by using a low-cost mini-computer-based system attached to an RGB camera module to monitor plant growth form top view. Environmental parameters were measured by different sensors, such as light intensity, soil humidity, and air temperature and humidity. The data was transferred via a Wi-Fi connection to a private IoT-Cloud gateway to apply further analysis.

3D images can be used in plant phenotyping as well. In the following some literature refers to the low-cost 3D plant phenotyping in a controlled environment will be explained. The first example is **l**ow-**c**ost RGB **p**henotyping lab (LCP lab) which was introduced in [36] that includes automated plant tracking using QR code, imaging setup, and image analysis stages. Figure 2.9 illustrates the image acquisition platform with its applications.

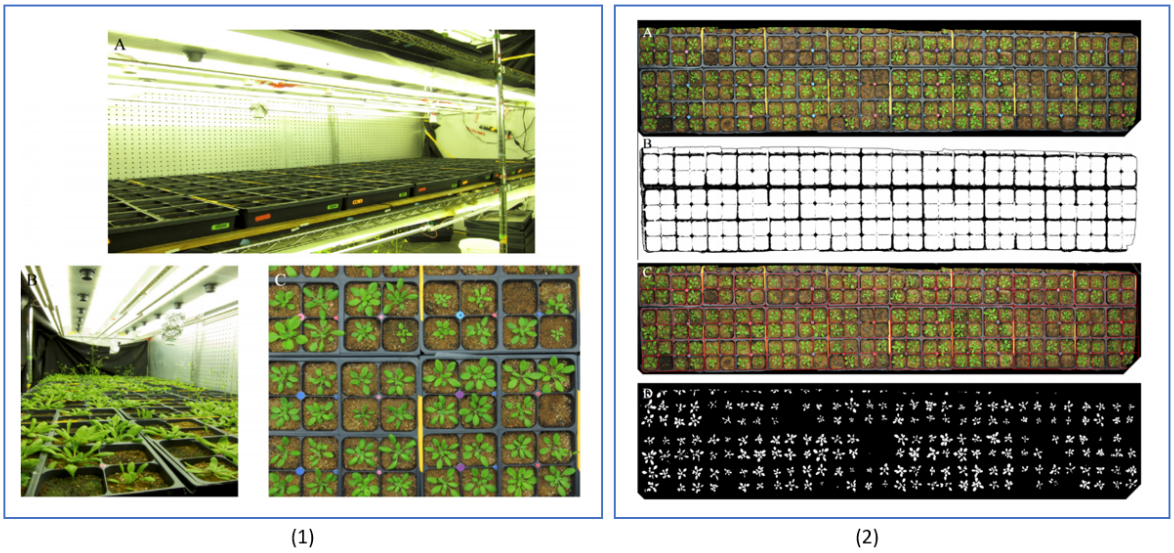


Figure 2.8 – (1) The low-cost indoor imaging platform. (2) Orthophoto processing. (A) The mosaicked orthophoto for half-shelf; (B) detected pot binary image; (C) generated 4-by-4 grid overlaid on the orthophoto; (D) detected plant binary image. Reproduced from [10].

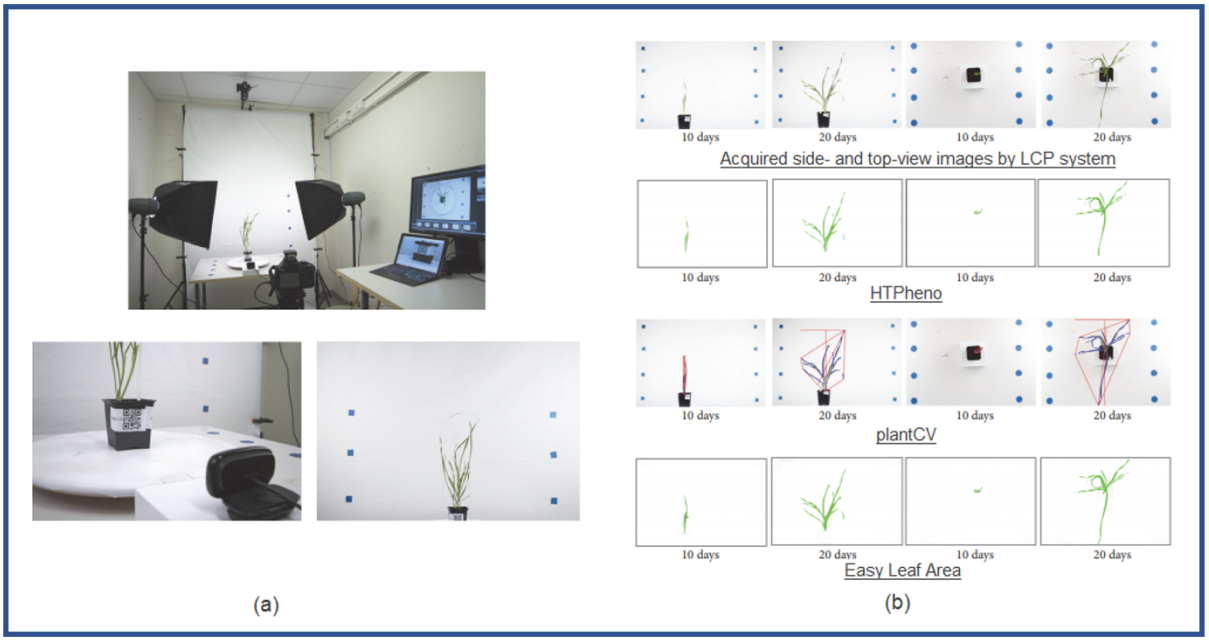


Figure 2.9 – (a) Low-cost RGB imaging phenotyping lab system, (b) Side- and top-view images of cultivar Nelson at three time-points after sowing processed by HTPheno [37], plantCV [38], or Easy Leaf Area [39]. Reproduced from [36].

In 2011, Microsoft Kinect for the first time was used as a low-cost RGB-D camera to 3D measurements on the shoot of entire plants in a controlled environment [1]. LiDARPheno [15] was presented as a low-cost LiDAR-based platform for phenotyping the plants in-lab and in-field. It consists of off-the-shelf, low-cost components and modules, including Arduino Uno, Raspberry Pi, servo motor-based mechanism, and a low-cost commercial 2D LiDAR scanning system. The depth vision-based plants phenotyping method in controlled environment, **V**ertical **F**arming with **A**rtificial **L**ighting (VFAL) was proposed in [12]. The method combines 3D plants modeling and deep segmentation of the higher leaves at the earlier growth stages, during a period of 25–30 days. Commercial close range RGB-D sensors are positioned on top of each tray, at the fixed distance. The plant height is computed by considering the depth map obtained for each tray Fig. 2.10 in different days after seeding (DAS). However, the plant surface provided both by the point cloud, and the depth map is complicated to distinguish individual plants, especially at later stages, at which only a carpet of leaves is visible. In the next step, both leaf area and leaf weight are computed by the first segmentation of leaves in visible layers from RGB images and then project them on the depth map.

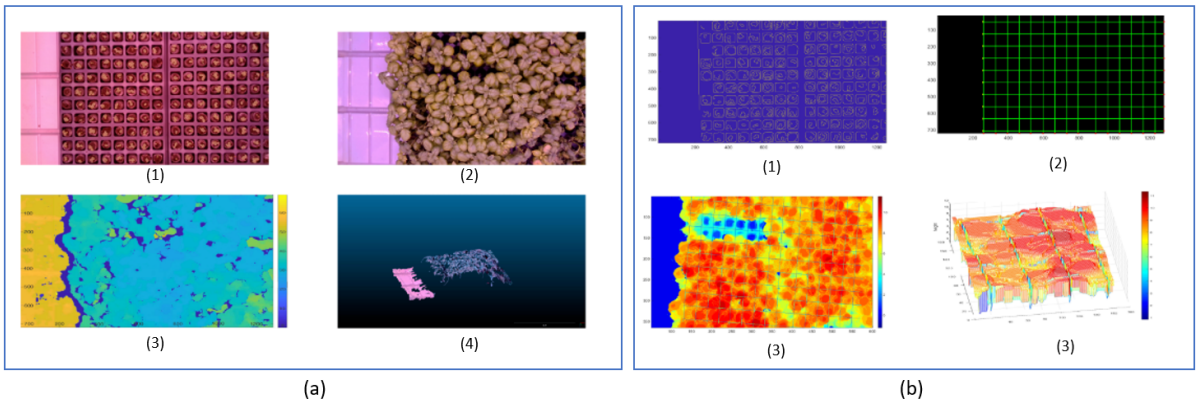


Figure 2.10 – (a1,a2) RGB images of a layer at DAS7 and DAS28, (a3) raw depth map from RGB-D sensor, (a4) Single point cloud from row depth map; (b1) Canny edge detection, with threshold 0.65. (b2) Lines fitted on the tray cells, with the Hough transform voting. (b3) Height surface at DAS24, in mm, (b4) 3D reconstructed leaves with the lines separating the tray cells visible, with height in mm. Reproduced from [12].

In another work, the diurnal pattern of leaf hyponasty and growth in the Arabidopsis plant was measured by using laser scanning in [40]. This study's objective was to measure the light-mediated growth responses in Arabidopsis and understand the underlying regulatory processes at cellular and molecular levels. The acquisition system is shown in

Fig. 2.11.

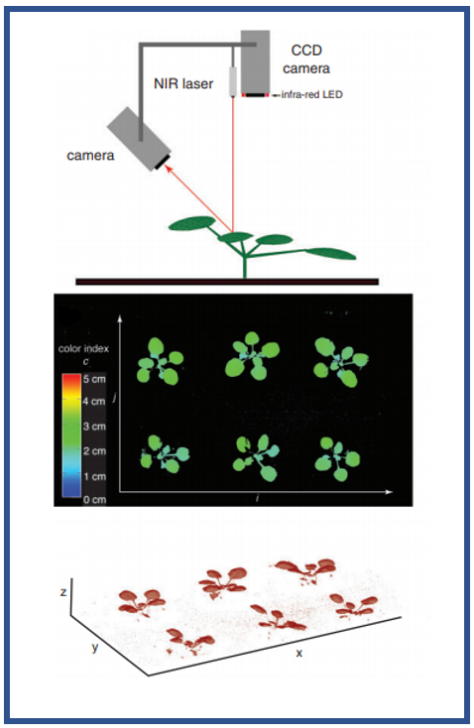


Figure 2.11 – Laser scanner and charged coupled device (CCD) camera mounted on the imaging unit in the Scanalyzer HTS phenotyping device and 2.5D height-scaled image in addition to the 3D point cloud computed from the recorded 2.5D image. Reproduced from [40].

A new cost-effective 3D imaging system was developed by [20], to collect images of soybeans plants from different viewpoints to reconstruct 3D models of the plant at the early growth stage. A low-cost digital camera mounted at a camera-arm controlled by a stepper motor driver was used to move the camera and take the plants' images from different viewpoints.

The importance of flexible and affordable plant phenotyping platforms is emphasized in [13] by developing an open-source and automated plant imaging system (PhenoBox) Fig. 2.12 and processing (PhenoPipe) solution that can be adapted to various phenotyping applications in plant biology and beyond for the evaluation of visual traits from plant shoot images. In the PhenoBox plant imaging system, the camera and turntable are controlled by a Raspberry Pi located in the electronics compartment in the lower right area of the system. PhenoBox system has broad applicability to study biotic and abiotic stresses in mono-cot and dicot species of varied sizes. The correlation achieved by the

affordable solution (PhenoBox/PhenoPipe system) was similar in strength to published results by highly cost 3D sensors.

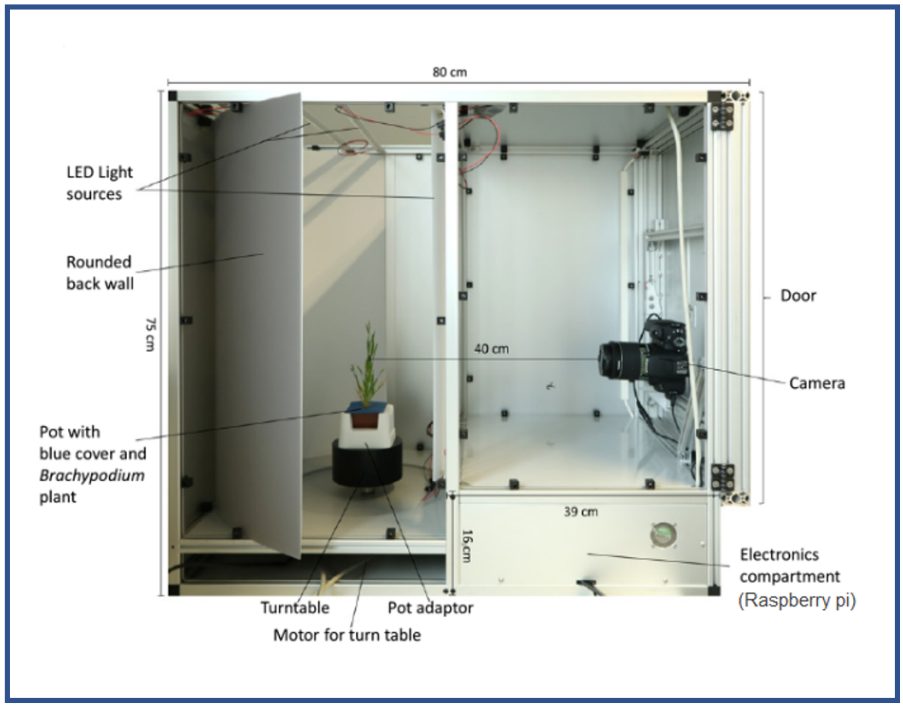


Figure 2.12 – Side-view of the PhenoBox. The camera and turntable are controlled by a Raspberry Pi 3 mini-computer. Reproduced from [13].

According to [16], the seedling growth reaction to seasonal change in the daytime was studied by using Raspberry Pi SBC connected to the infrared-sensitive camera. In this study, the Arabidopsis seedlings were monitored individually by time-lapse imaging in different light conditions.

An image capturing system introduced by [14], which consists of a near-infrared LED panel with a NoIR Raspberry Pi camera mounted to a mini-computer. A MatLab-based software module (iDIEL Plant) was developed to characterize Rosette's expansion. Plants were imaged approximately three weeks after germination every 20 min throughout the 24h light-dark growth cycle. The result provided a dynamic and uninterrupted characterization of differences in Rosette growth and expansion rates over time for the three lines tested. The described image acquisition system is shown in Fig.2.13.

All this mentioned literature depicted heretofore emphasizes the strength of single-board computers and low-cost imaging sensors due to the affordability, reliability, and flexibility for developing low-cost image-based phenotyping platforms. In this thesis we

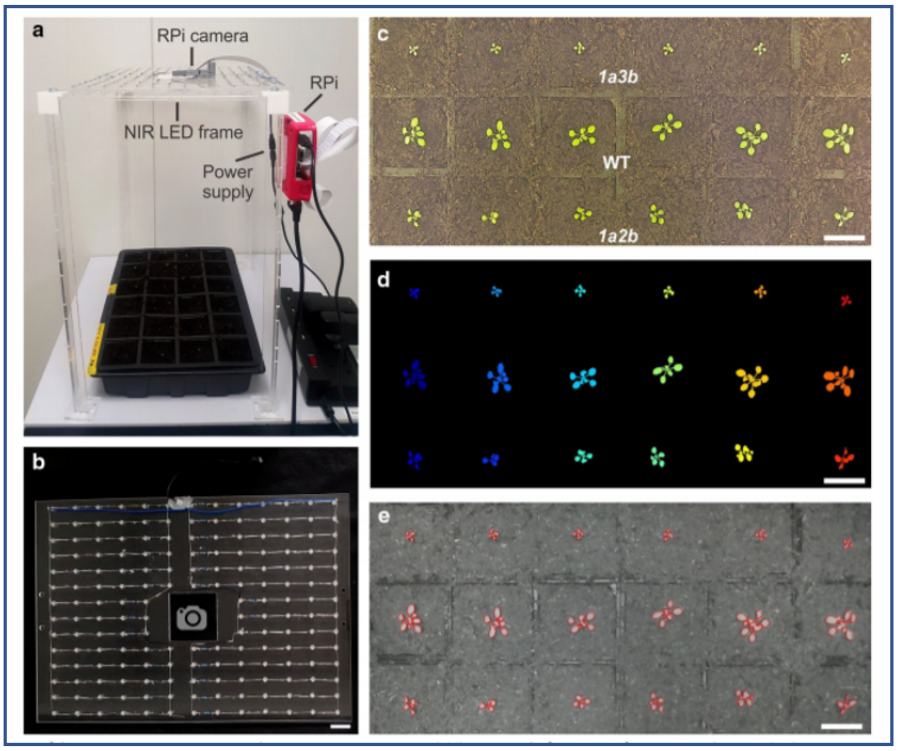


Figure 2.13 – (a) Image capturing system includes a NIR LED frame, and a NoIR RPi camera mounted to a Raspberry Pi mini-computer. (b) The NIR LED frame consists of 173 NIR LEDs arranged in parallel circuits. (c) plants under visible light (VIS) conditions. (d) Automated segmentation of plants. (e) Near-infrared (NIR) image of plants shown in (c) taken in the dark illuminated by NIR LEDs. Reproduced from [14].

used these existing technologies and deployed them in grid of sensors for original use cases. In the following sections, we describe our contributions to the low-cost imaging of seedling growth by monitoring seedling growth in both canopy and individual observation scale.

2.3 Contributions to low-cost imaging of seedling growth

As stressed in the introduction section, a favorable scenario for low-cost plant imaging comes with the use of grid of sensors monitoring from top view plants at their early stages of development. In this section, we address two plant phenotyping use cases with this approach while considering two observation scales. First, we investigate the use of low-cost time-lapse RGB imaging systems in a single pot observation scale for documenting plant development at the seedling level, where plants have simple architectures and are

not touching or self-occluding themselves. Next, we push forward and study plant growth in the presence of plant touching or self-occluding themselves.

2.3.1 Seedling growth monitoring in individual observation scale

A specificity of plants is their continuous capability to metamorphose during their lifetime. This process is characterized by the kinetics of ontological development stages, i.e., stages that occur in a definite order. In this scientific study, we focus on some of these connected steps of a plant's life at the seedling level. The period from seed germination in the soil to the development of the first true leaf is crucial for the plant. During this time, the seedling must determine the appropriate mode of action based on its environment to best achieve photosynthetic success and enable the plant to complete its life cycle. Once the seedling emerges out the soil, it initiates photomorphogenes is, a complex sequence of light-induced developmental and growth events leading to a fully functional leaf. This sequence includes severe reduction of hypocotyl growth, the opening of cotyledons, initiation of photosynthesis, and activation of the meristem at the shoot apex, a reservoir of undifferentiated cells that will lead to the formation of the first leaf [41]. The molecular mechanisms regulating these time-based events involves profound reprogramming of the genome that is challenging to study in field situation because the heterogeneity of the seedling population must be taken into account. It is essential to understand this seedling development process from an agronomic point of view because the seedling establishment is critical to crop yield. Uneven emergence timing, for instance, is associated with lower yields and poor farmer acceptance.

In this context, time-lapse imaging is a valuable tool, accessible at a rather low-cost [42, 6, 4, 43], for documenting plant development and can reveal differences that would not be apparent from a sole endpoint analysis. At the seedling level where plants have simple architectures, such time-lapse imaging can be done from top view to provide an efficient solution for seedling vigor assessments and monitoring of seedling growth. While some statistical tools transferred from developmental biology exists to perform time-to-event analysis [44], a current bottleneck [3] lay in the automation of the image analysis. A recent revolution occurred in the field of automated image analysis with deep neural networks [45], which have shown their universal capability to address almost any image processing challenges with high accuracy. This revolution also benefits plant imaging [46], and it is currently a timely topic to adapt these tools, which came from the artificial intelligence community to specific topics of interest in plant sciences. In this study, we

propose an entire pipeline based on deep learning dedicated to the monitoring of seedling growth.

Seedling monitoring with computer vision has received considerable attention in the literature including [47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59]. Several studies consider germination and seedling growth measurements in vitro, using plastic boxes or paper towel [47, 48, 49, 50, 51, 52, 53, 56], which enable the monitoring of radicle emergence (germination) or organ growth (seedling growth). Others, like in this work, used soil-based sowing systems, where seedling emergence and early developmental events of the aerial part can be determined under more realistic agronomical conditions [55, 58, 57, 59, 60, 61].

Reported approaches to monitor seedling from the top view in the soil are effective for a large set of crops, mainly at the emergence level, i.e., seedling counting to determine stand establishment [55, 58, 59, 60, 61], or estimating early plant vigor by spectral imaging or measuring the leaf area index of the small plants [61, 55, 57]. Here we propose to push forward the detection of the early seedling developmental stages to be able to monitor the kinetics of early seedling development in the soil from cotyledon emergence until the development of the first real leaf. We propose to tackle this task, for the first time to the best of our knowledge, with a deep learning-based approach. While, as most related work, deep learning has been applied to the problem of seedling detection and segmentation [59] as well as detection of wheat spikes [62], this has been performed at a fixed stage of development. In another similar work [42], a graph-based method for detection and tracking of tobacco leaves at the late stage of the plant growth from infrared image sequences was proposed, where all tobacco plants used in the experiments were of the same genotype. In the last similar approach [63], a feature-based machine learning algorithm was developed to detect two stages of heading and flowering of wheat growth. In our study, we specifically investigate, how the existing methods of deep learning, can incorporate time-dependency in sequences of images to solve a problem of developmental biology such as the one of seedling development.

The proposed plant method includes five main items: (a) The imaging system developed to create (b) the dataset, which needs to benefit from (c) pre-processing before investigating (d) various approaches for the detection of developmental stages of seedling growth based on deep learning methods and (e) post-processing.

Image acquisition platform by Raspberry Pi and RGB camera

We installed 60 Raspberry Pi 3B connected to RGB PiCamera modules with a spatial resolution of 3280 by 2464 pixels to image seedlings from the top view, as illustrated in Fig. 2.14. The distance of 50 cm was chosen to allow the observation of 2 trays of 200 pots per camera. To setup consistent image acquisition platforms, the imaging system must be configured across long experiments and data compares from multiple Raspberry Pi camera rigs. To acquire this stable configuration, we used commercial Raspberry Pi cases to wrap the board of the computer and suspend it to the roof of each shelf. For this top-down imaging system, an AC multi-socket plugin has considered for each three Raspberry Pis where all these multi-socket connected to an uninterruptible power source (UPS) in order to avoid any dis-connectivity of electricity. Time-lapse imaging was scheduled at 15-minute (can be changed based on the purposes) intervals using a python script. Images were pulled from each Raspberry Pi to a server after capturing each image by server-side scripts using SSH.

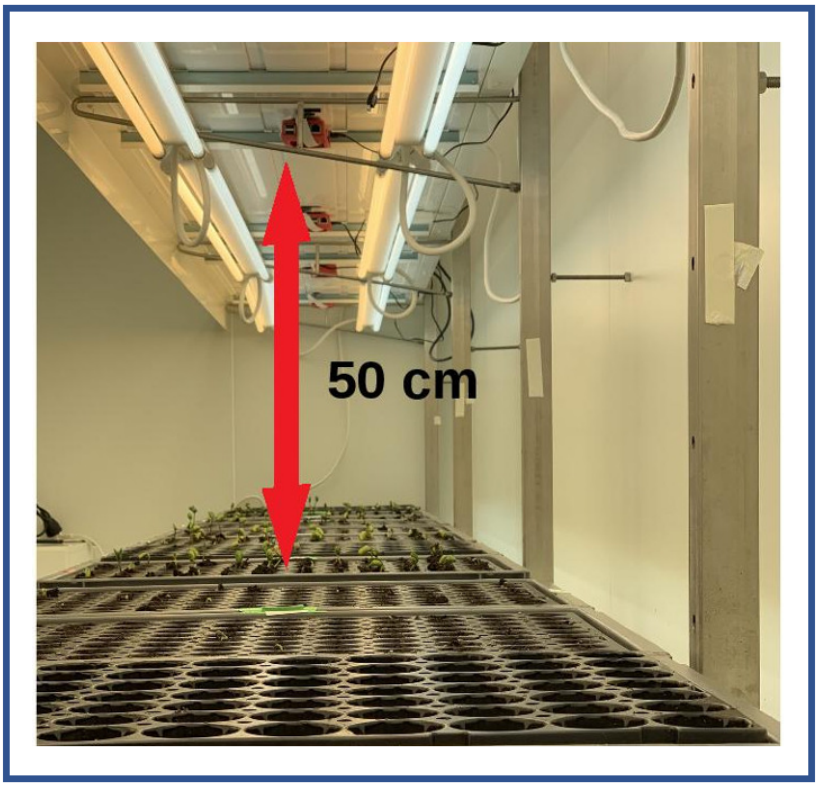


Figure 2.14 – Imaging system installed in a growth chamber.

	Species	No. of Trays	No. of Pots in each tray	No. of Temporal sequences	Total No. of images
Training dataset	Red clover	2	200	400	307,200
Validation dataset	Red clover	1	200	200	153.600
Testing dataset	alfaalfa	2	200	400	307,200

Table 2.4 – Description of the split of the annotated data set for training models.

Dataset

Seedling establishment was recorded for 3 experiments using seed lots from different accessions of red clover (Trifolium pratense) (experiment 1) and alfalfa (Medicago sativa) (experiments 2 and 3). Each experiment consisted of 70 trays with 200 pots in which 50 seeds of four accessions were sown. Soil pots were hydrated to saturation for 24h, after which excess water was removed. After 24h, seeds were sown at a depth of 2 cm, and trays were placed in a growth chamber at 20°C/16°C, with 16h for photoperiod at $200\mu Mm^{-2}s^{-2}$. The soil was kept humid throughout the experiment.

Each experiment took two weeks with a time-lapse of 15 minutes. In total, the database consists of 42000 temporal sequences of RGB images of size $89 \times 89 \times 3$ pixels where each temporal sequence consists of 768 individual images. During day time, images were captured while images were automatically discarded during night times due to the absence of illumination. An example of images from the database is shown in Fig. 2.15. Among all temporal sequences, images of 3 randomly selected trays were annotated from the first experiment (red clover species) and 2 trays from the second experiment (alfalfa species). Annotation consisted of four classes: soil, the first appearance of the cotyledon (FA), the opening of the cotyledon (OC), and the appearance of the first leaf (FL). To avoid cross sampling, we considered images of the red clover trays for training (two trays) and validation (one tray) datasets. The testing dataset consisted of images of the remaining two trays from the alfalfa. Table 2.4 provides a synthetic view of the data set used for training and testing of the models.

Pre-processing

Raw images were then sent to pre-processing before being applied to the deep learning method investigated in this study. A filtered variant of the raw images was also created where the soil background was removed from images. This filter was produced by applying a color filter on images in the HSV color domain to keep the green range of images in the Hue channel. This strategy was found robust because the soil used during the experiment

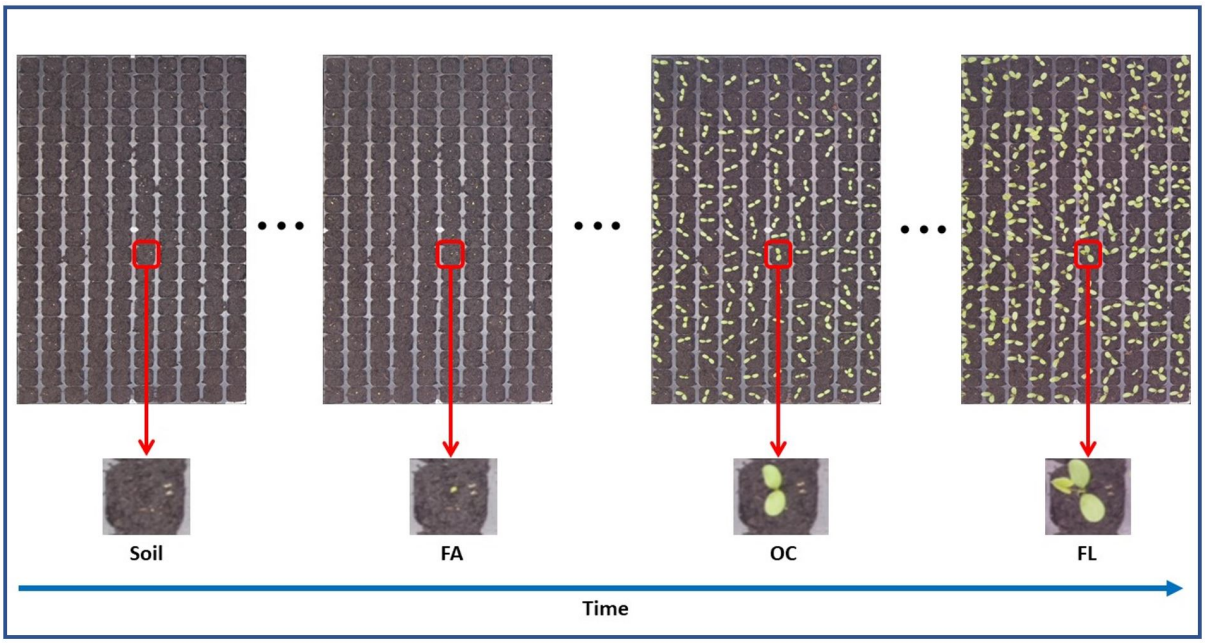


Figure 2.15 – An overview of the time-lapse collected for this work. Upper row, view of a full tray with 200 pots from the top view. Lower row, a zoom on a single pot at each stage of development to be detected from left to right: soil, the first appearance of the cotyledon (FA), opening the cotyledons (OC) and appearance of the first leaf (FL).

was the same, and that lighting was kept constant. Figure 2.16 shows an example of images with and without background. To being low-cost, we configured the time-lapse

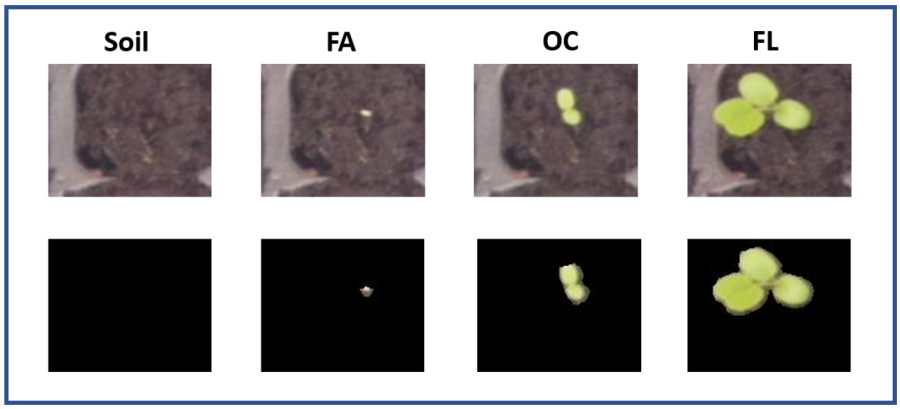


Figure 2.16 – Two different types of data used in training and testing. Up: Original images, Down: Images without background

imaging system with two different sensors in the canopy observation scale. However, as described in the previous section, since we need to study specific seedlings' specific traits,

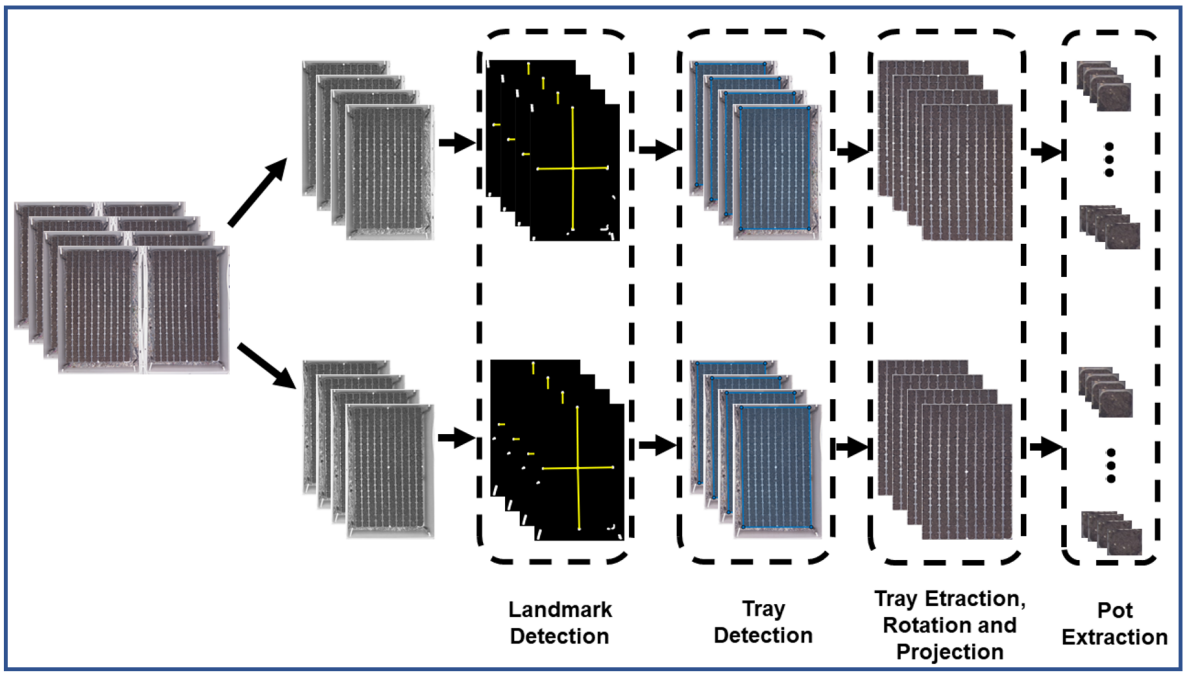


Figure 2.17 – Pot extraction workflow.

the pre-processing applied to provide an efficient solution for seedling vigor assessments and monitoring of seedling growth individually.

Since deep learning methods have to predict the seedling developmental stage on an individual basis, the raw images of Fig. 2.15 could not be directly applied to the neural networks. Thus, the first step of pre-processing was to extract produced crops of each pot. To extract them, we needed first to detect, extract, and adjust trays; then, pots were extracted from trays. Figure 2.17 shows a workflow of the pot extraction from trays, which includes three steps described below.

Landmark detection: In this experiment, trays used included five white landmarks located at the center and four corners of the trays. Because of the constant control of lighting conditions, these five landmarks were detected with a fixed threshold. Then, the five most prominent objects were kept, and the possible remaining small objects were removed. Among the five significant landmarks, the most central object in the images was considered as the central landmark. At the next steps, the four other landmarks were detected based on their minimum angle corresponding to the central landmark with horizontal and vertical axes.

Tray detection and extraction: In this step, coordinates of the trays were detected using the landmarks. Then, based on the coordinates of these landmarks, trays could be

extracted from the image. Since trays may not be positioned precisely along the axis of the vertical and horizontal axis sensor of the camera, the trays need to be rotated. The orientation of the trays was found after the computation of the first eigenvector's angle in the principal component analysis of the Fourier transform modulus [64]. Finally, a geometric transformation algorithm [65] was implemented to project the rotated trays to make them straight.

Pot extraction: In the last step, all 200 pots of each tray were extracted as an independent temporal sequence of images by using a sliding window with a stride of one pot. The size of these sliding windows was made adjustable by the user to fit with the size of the pot.

This pre-processing pipeline of Fig. 2.17 has some generic value. Since we did not find something equivalent in the literature for our purpose, we decided to make it available as supplementary material under the form of a free executable [3]. We believe that this can be used as a useful tool for any imaging of traits despite the simplicity of principle.

Deep learning methods

The three plant events plus soil (Soil, FA, OC, and FL) were expected to occur in a definite order. Different strategies to take benefit from this ontological prior-knowledge on the development were tested and described in the following subsection.

Baseline 4-class CNN

As a naive baseline approach, we designed a convolutional neural network (CNN) architecture to predict the classes of each event of Soil, FA, OC, and FL from each frame of the time-lapses independently and without any additional information regarding the temporal order in which they should occur. Given a training set including K pairs of images x_i and labels $\hat{y}_i$, we trained the parameters θ of the network f using stochastic gradient descent to minimize empirical risk

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{K} \mathcal{L}(\hat{y}_i, f(x_i, \theta)) \tag{2.1}$$

where $\mathcal{L}$ denotes the loss function, which was chosen as cross-entropy in our case. The minimization was carried out using the ADAM optimizer [66] with a learning rate of 0.001.

3. `https://uabox.univ-angers.fr/index.php/s/HJAHp0bhZv1zy1j`

Our proposed architecture $f(\cdot,\cdot)$, shown in Fig. 2.18, consisted of two main blocks, the feature extraction block, followed by a classification block. In a CNN model, the feature extraction block takes care of extracting features from input images by convolutional layers, and the classification block decides classes. The proposed CNN architecture has been optimized on a hold-out set. It is given as follows: four convolutional layers with filters of size 3×3 and respective numbers of filters 64, 128, 256, and 256 each followed by rectified linear unit (RelU) activations and 2×2 max-pooling; a fully connected layer with 512 units, ReLU activation and dropout (p=0.5) and a fully connected output layer for four classes corresponding to each event with a softmax activation.

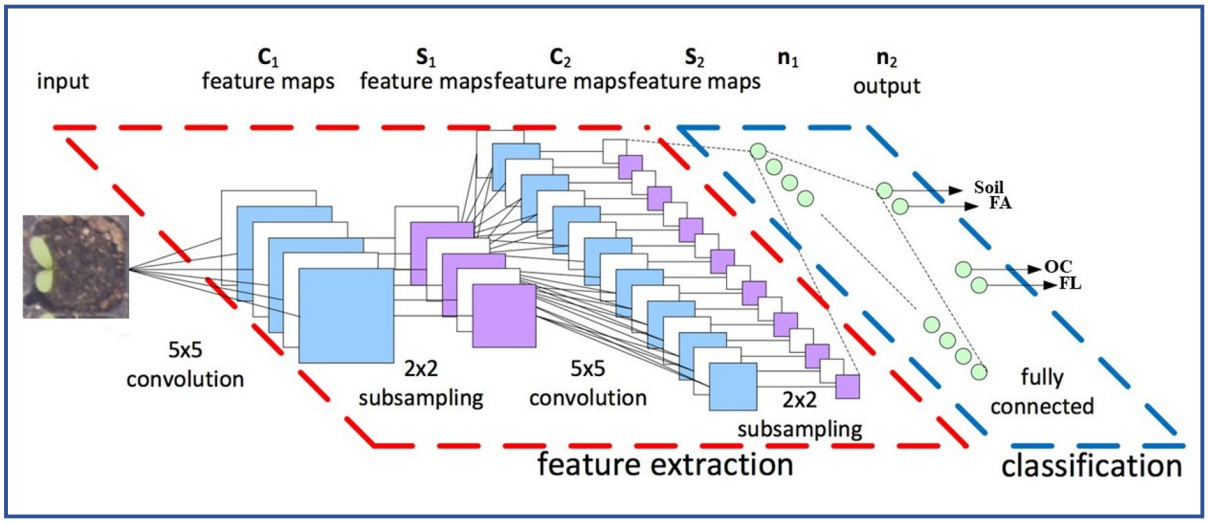


Figure 2.18 – CNN architecture designed to serve as baseline method for the independent classification of each frame of the time-lapses into one of the three stages of plant growth plus soil (Soil, FA, OC, and FL) without any prior temporal order information.

2-class CNN's

The baseline 4-class CNN architecture illustrated in Fig. 2.18 is naive because it does not incorporate the prior knowledge of the ontology of plant growth to decide between different growth steps of plants plus soil (Soil, FA, OC, and FL). As a first improvement of the previous naive baseline, we implemented a variant of the CNN model of Fig. 2.18 dedicated to the binary classification of two consecutive stages of development. We thus trained 3 models detecting between M_1(Soil, FA), M_2(FA,OC) and M_3(OC,FL). At the beginning of the analysis of an entire time-lapse sequence M_1 is used. When a first FA is detected M_2 is applied, and so on until the first FL detection is reached.

CNN followed by Long short-term memory

The 2-class CNN includes the prior knowledge of the ordered development of the seedling along with a given ontology. However, this prior knowledge is added on top of the CNN. In order to bring a memory directly inside the CNN model, the Long-Short Term Memory (LSTM) [67, 68] architecture was embedded between the feature extraction block and the classification block of the proposed CNN model. LSTM as a special RNN structure has proven stable and powerful for long-range modeling dependencies in various previous studies [68, 69, 70]. The major innovation of LSTM is its memory cell c^t, which essentially acts as an accumulator of the state information. The cell is accessed, written, and cleared by several self-parameterized controlling gates. Whenever a new input comes, its information will be accumulated to the cell if the input gate i^t is activated. Also, the prior cell status c^{t-1} could be « forgotten » in this process if the forget gate f^t is on. Whether the latest cell output c^t will be propagated to the final state h^t is further controlled by the output gate o^t. One advantage of using the memory cell and gates to control information flow is that the gradient will be trapped in the cell [68] and be prevented from vanishing too quickly. In a multivariate LSTM structure, the input, cell output, and states are all $1D$ vectors features from the feature extraction block of the proposed CNN model. The activations of the memory cell and three gates are given as

$$\begin{aligned} i^t &= \sigma(W_{xi}x^t + W_{hi}h^{t-1} + W_{ci}c^{t-1} + b_i) \\ f^t &= \sigma(W_{xf}x^t + W_{hf}h^{t-1} + W_{cf}c^{t-1} + b_f) \\ c^t &= f^t c^{t-1} + i^t tanh(W_{xc}x^t + W_{hc}h^{t-1} + b_c) \\ o^t &= \sigma(W_{xo}x^t + W_{ho}h^{t-1} + W_{co}c^{t-1} + b_o) \\ h^t &= o^t tanh(c^t) \end{aligned} \tag{2.2}$$

where $\sigma()$ is the sigmoid function, all the matrices W are the connection weights between two units, and $x = (x^0, ..., x^{T-1})$ represents the given input.

The CNN-LSTM architecture is an integration of a CNN (Convolutional layers) with an LSTM. First, the CNN part of the model process the data and extract features then the one-dimensional feature vectors feed to an LSTM model to support sequence prediction. CNN-LSTMs are a class of models that is both spatially and temporally deep and has the flexibility to be applied to a variety of vision tasks involving sequential inputs and outputs. Fig. 2.19 shows a schematic of a CNN-LSTM model. The proposed CNN-LSTM model consisted of the same convolutional layers as the 4-class CNN model of Fig.2.18

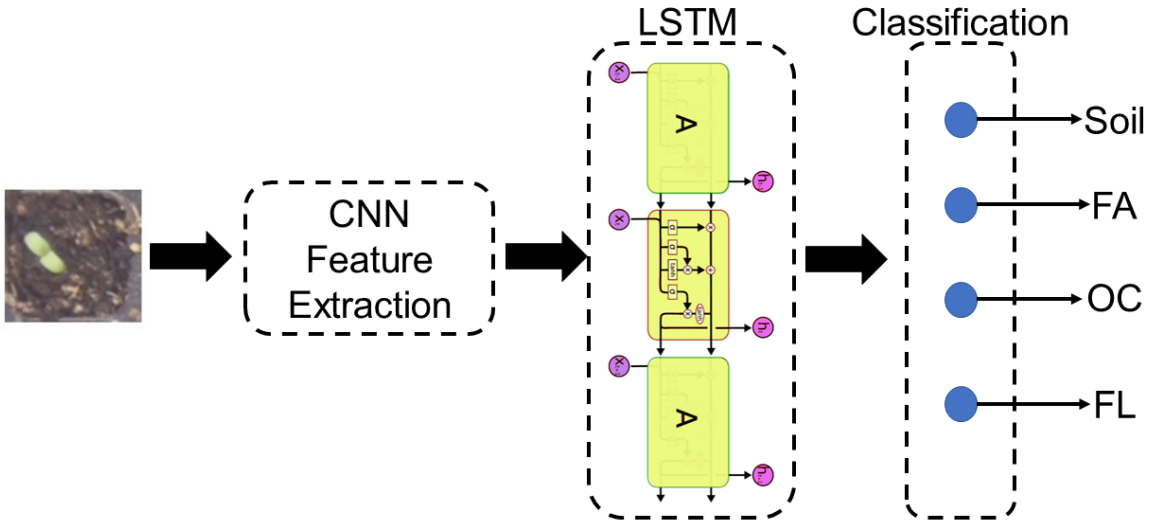


Figure 2.19 – CNN-LSTM block.

and an LSTM layer with 128 units.

Convolutional LSTM (ConvLSTM)

As an alternative to CNN-LSTM, we use ConvLSTM [71] which has convolutional structures in both the input-to-state and state-to-state transitions. In ConvLSTM all the inputs $X^1;\cdots;X^t$, cell outputs $C^1;\cdots;C^t$, hidden states $H^1;\cdots;H^t$, and gates i^t; f^t; o^t of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). The ConvLSTM determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and input-to-state transitions. The key equations of ConvLSTM are shown in Eq. (2.3) below, where '$\circledast$' denotes the convolution operator. Figure 2.20 shows a schematic of the ConvLSTM method adopted for our

purposes.

$$\begin{aligned} i^t &= \sigma(W_{xi} \circledast x^t + W_{hi} \circledast h^{t-1} + W_{ci} c^{t-1} + b_i) \\ f^t &= \sigma(W_{xf} \circledast x^t + W_{hf} \circledast h^{t-1} + W_{cf} c^{t-1} + b_f) \\ c^t &= f^t c^{t-1} + i^t tanh(W_{xc} \circledast x^t + W_{hc} \circledast h^{t-1} + b_c) \\ o^t &= \sigma(W_{xo} \circledast x^t + W_{ho} \circledast h^{t-1} + W_{co} c^{t-1} + b_o) \\ h^t &= o^t tanh(c^t) \end{aligned} \tag{2.3}$$

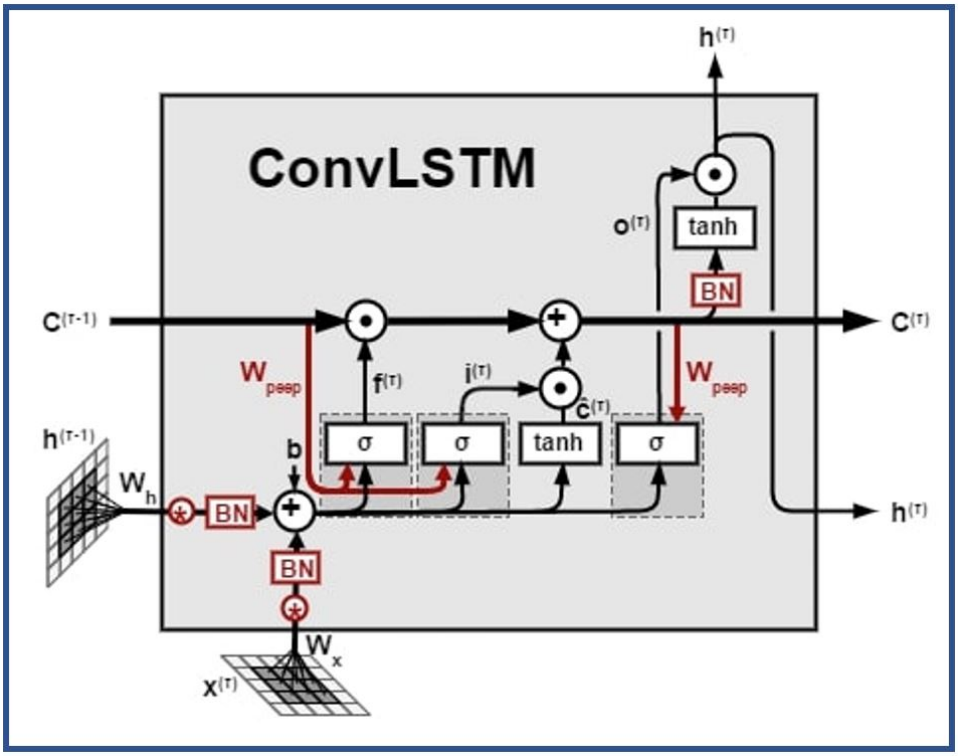


Figure 2.20 – ConvLSTM block with one cell [71]

Post-processing

The passing from one developmental stage to another can consist of very tiny details. This was, for instance, the case for FA and FL in our case. Filtering was applied to the classified data to denoise them. This filter illustrated in Fig. 2.21, was based on a sliding window computing a majority voting by finding the median of classes(2.4)

$$c = \left\lfloor \left\{ \left(\frac{n+1}{2} \right) \right\}^{th} \right\rfloor \tag{2.4}$$

where c and n represent predicted class and window size, respectively.

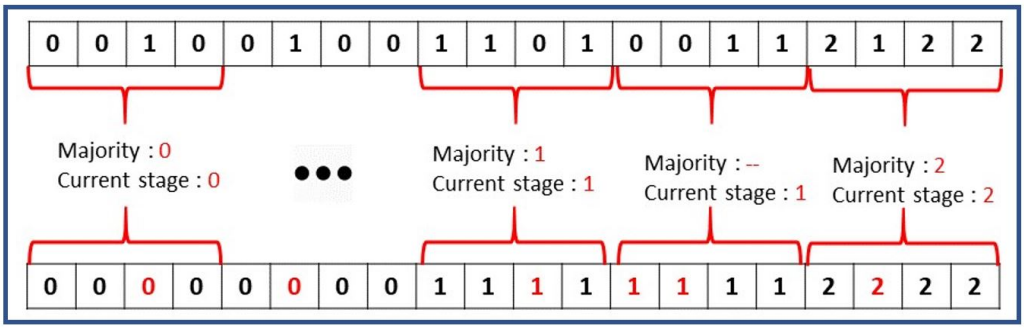


Figure 2.21 – An example of the post-processing step on predicted classes where the sliding window size is four images.

Additionally, this window replaced all neighbors' current stage to all labels that were detected as the previous stage. The size of the sliding window was optimized on the CNN-LSTM and 4-class CNN architecture. As shown in Fig. 2.22, performances were found optimal for both architectures on the training data set for a size of 4 frames, corresponding to an observation of 1 hour in our case.

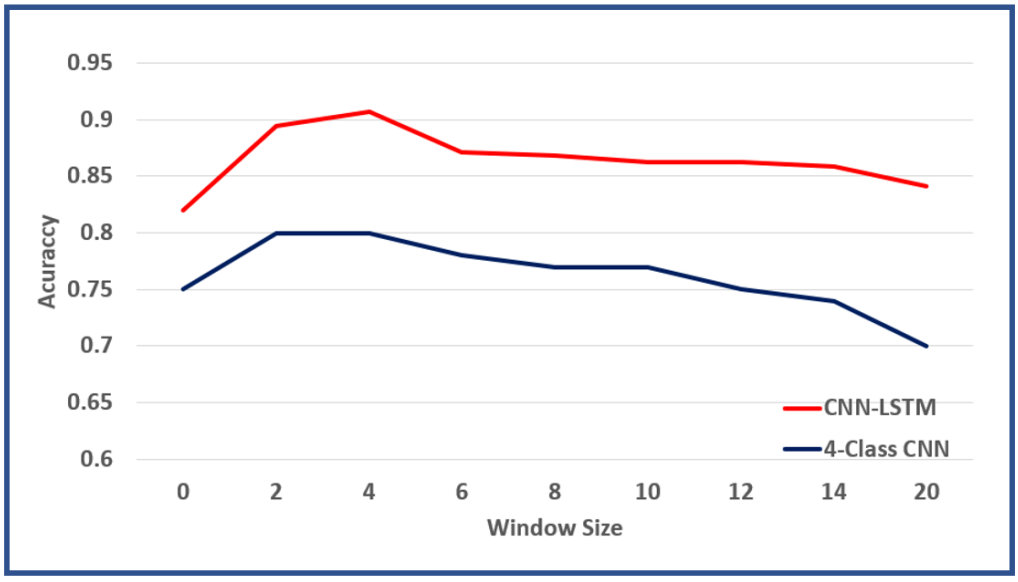


Figure 2.22 – Classification accuracy as a function of denoising windows size.

Results and discussion

The proposed deep learning methods 4-class CNN, 2-class CNN's, CNN-LSTM, and ConvLSTM were applied to the dataset produced by our imaging system after pre-processing and post-processing as described in the previous section. We now present and discuss the associated results. The performances of the different deep learning methods tested on our dataset were assessed with classical metrics such as accuracy, error, sensitivity, specificity, precision, and false alarm positive rate. They are provided in Tables 2.5 and 2.6, respectively, for images with and without soil background.

Tables 2.5 and 2.6 show that all methods performed better than the naive 4-class CNN architecture, which was processing the temporal frames independently of any prior knowledge on the order of the ontological development of seedling. The best strategy to incorporate this knowledge among the one tested was found to be the CNN-LSTM architecture, which outperforms all other models for all tested metrics. Removing the soil numerically, clearly improves all methods while keeping the CNN-LSTM architecture as the best approach.

Model	Accuracy	Error	Sensitivity	Specificity	Precision	FalsePositiveRate
4-class CNN	0.63±0.20	0.37±0.20	0.63±0.2	0.94±0.05	0.88±0.1	0.06±0.05
2-class CNN's	0.72±0.25	0.28±0.26	0.72±0.24	0.95±0.06	0.90±0.11	0.08±0.05
CNN-LSTM	0.83±0.10	0.15±0.10	0.82±0.10	0.93±0.06	0.85±0.10	0.06±0.06
ConvLSTM	0.62±0.2	0.33±0.2	0.68±0.2	0.93±0.07	0.84±0.1	0.06±0.06

Table 2.5 – The average performance of models with different evaluation metrics on images with soil background.

Model	Accuracy	Error	Sensitivity	Specificity	Precision	FalsePositiveRate
4-class CNN	0.80±0.19	0.20±0.19	0.85±0.13	0.93±0.07	0.85±0.14	0.07±0.07
2-class CNN's	0.88±0.18	0.12±0.18	0.86±0.10	0.95±0.05	0.86±0.11	0.05±0.05
CNN-LSTM	0.90±0.08	0.10±0.0.07	0.87±0.11	0.96±0.03	0.88±0.15	0.04±0.04
ConvLSTM	0.81±0.11	0.21±0.09	0.85±0.03	0.92±0.09	0.85±0.12	0.07±0.10

Table 2.6 – Average performance of models on images without soil background.

Our experimental results show that a reasonable recognition rate of plant growth stages detection (approximately 90%) can be achievable by the CNN-LSTM model. It is possible to have a more in-depth analysis of the remaining errors by looking at the confusion matrix of this CNN-LSTM model, as given in Table 2.7. This confusion matrix shows that most of the errors, almost 98%, happen between the most complicated classes

of OC and FL while the remaining 2% of errors appear on borders of the first two classes of soil and FA.

		Predicted			
		Soil	**FA**	**OC**	**FL**
True Classes	**Soil**	**97531**	0	0	0
	FA	2591	**26855**	2915	0
	OC	0	0	**58668**	19556
	FL	0	0	8219	**90610**

Table 2.7 – Confusion matrix of cross-subject performance where the best deep learning method, the CNN-LSTM architecture is used.

One may wonder where the classification errors in this experiment can come from. In our error analyses, we found four different sources of errors in the experiment. The first source of errors can come from the different cotyledons and leaf sizes of the two species, as the cotyledons and leaf size of a species can be much bigger or smaller compared with other species. Usually, this type of error happens in the borders of two classes of OC and FL. Figure 2.23 shows an example of these differences in the size of two plant species. Data augmentation with a variation on the zoom could be a solution to help with these errors. The second source of errors can be due to the circadian cycle of plants during the

Figure 2.23 – A sample of images from two plant species used for training (left) and testing (right) dataset

growth. The circadian cycle of plants makes some movements on cotyledon and leaves during day and nights [72]. This type of error can happen at the border of FA and OC, where these movements make a delay for the detection of fully opening cotyledon. Also, this type of error can happen at the border of two classes of OC and FL, where the circadian cycle does not allow the system to recognize the appearance of the first leaf from the middle of the cotyledon.

The third source of errors happens due to the overlapping of plants in a tray. Plants grow at different speeds and directions in a tray, and it makes overlapping on plants of neighbor pots at some points. This type of error usually happens in the last two classes of OC and FL.

The last source of the errors can come from annotation errors. In general, the annotation of plant growth stages is challenging since plants grow continuously; it means there are no striking events of growth. In this case, a class represents a period of growth. For instance, the FA class is assigned to images which are capture in the period of the first appearance of the cotyledon till the time of the fully opening of the cotyledon. In this case of annotation, different annotators may define the ending of a stage period with an approximate delay of 15 images. Also, there is a period of formation of the first leaf before its unfolding during plant growth. This period is considered to be a part of the FL class in this experiment. This consideration may bring an additional error for annotation of stages as different annotators may recognize the beginning of the leaf formation with a delay.

Conclusion and perspectives

In this section, we have presented a complete imaging,image processing and machine learning pipeline to classify three stages of plantlet growth plus soil on the different accessions of two species of red clover and alfalfa.

Different strategies were compared in order to incorporate the prior information of the order in which the different stages of the development occur. The best classification performance on these types of images was found with our proposed CNN-LSTM model, which achieved 90% accuracy of detection with the help of a denoising algorithm incorporating the ontological order in the development stages.

These results can now be extended in various directions. It will be interesting to extend the approach to a range of species of agricultural interest in order to provide a library of trained networks. From this perspective, it could be interesting to investigate quantitatively how, by their similarity in shape, the knowledge learned on some species could be transferred to others via transfer learning, domain adaptation, or hierarchical multi-label classification [73].

More events of the development of plants could also be added to extend the investigation of seedling kinetics. This includes for instance the instant where cotyledons are out of soil fully or rise of the first leaf before unfolding. These extensions could be tested

easily following the global methodology presented in this section to assess the deep learning models. As another direction in this section, since we used classical standard RGB images, plants were not measured during nights, and some missed events could shift the estimation of the developmental stages of the seedlings. LiDAR cameras, accessible at low-cost [1], could be used to access to night events. This is what will be investigated in the following section.

Last, for even more advanced stages of development and yet still accessible from top view, the issue of plants overlapping each other would arise and become a limitation. Solving this would require to switch to tracking algorithms in order follow and label the trajectory of each plant despite ambiguity created by partial occlusion and overlapping. Other deep learning architectures would have to be tested in this perspective [74]. Another approach to solve such issue is to consider the surface constituted by the groups of touching plants (canopy) as a texture from which information can be extracted.

In this section we have monitored the development of individual plantlets not touching nor overlapping themselves during day time only. In the following section we investigate populations of plantlets touching and overlapping monitored during day and night.

2.3.2 Seedling growth monitoring in canopy observation scale

In this section, we consider the complex situation in which a population of plants possibly touching each others is to be monitored for quantifying their growth. Instead of developing an algorithm to segment each plant as in the previous study, we consider the 2.5D surface formed by the canopy of the plant as a whole. Under the assumption of stationarity of the growth pattern from one plant to another, we consider the average distance of this canopy to the camera as a signal characterizing the average population's global growth. This work proposes a signal processing analysis of the plant growth process.

In comparison with the closest related work [75], we use very low-cost imaging systems (hundred euros versus keuros) while observing larger populations (hundred of plants versus ten plants), over a longer time scale including the appearance of new leaves (two weeks versus one week). We demonstrate that this growth signal analysis can be used to recognize growth anomalies (while only controlled plants were exhibited in [75]). This is obtained with simple Fourier series, while more advanced wavelet analysis was used in [75].

In the following, we demonstrate that the situation can be understood as a signal processing problem. In addition to the instantaneous growth rate, other traits linked with

growth appear under the form of periodic patterns that we analyze in the Fourier domain. We identify the cause of these periodic patterns and show their value for agronomic applications.

Database

As shown in Fig. 2.24, we positioned 13 Kinects v.2 connected to LattePanda mini-computers gazing from the top view on populations of tomato's seedlings automatically irrigated. Per each sensor, 100 plants are captured, which means that this imaging system is high throughput. Using Kinect allows us to capture depth images during night times despite the absence of illumination. The plants were observed during two weeks with the time-lapse of 15 minutes, similarly to the previous research, after which they tend to bend, and the distance to the camera no longer precisely corresponds to their actual height. However, monitoring over these two weeks is of high relevance since they correspond to the first two weeks of the plant life after emergence from the seed. This is a stage of interest for biologists since photosynthesis is activated and a stage of interest for breeders since this is where they sell their products. This early stage is often crucial to the prognostic of the full plant development and its yield. The plants monitored here were seedlings of tomato. However, the approach can equally be applied to any species of interest.

So far the material produced here can already be used for educational purposes to provide the students with new application fields of the Fourier analysis. To this purpose we give access to raw data of our experiment [4].

Pre-processing

The produced depth map is converted after sphericity correction into a distance map of the population of plants to the camera. The depth imaging system used is an active imaging system based on infrared lighting; it, therefore, enables to monitor plant growth during the night. The depth map is thresholded to remove the soil. The average value of this distance map is computed and plotted as a function of time with a time-lapse of one image every 15 minutes.

4. `https://uabox.univ-angers.fr/index.php/s/hf2csYRguVKNtWy`

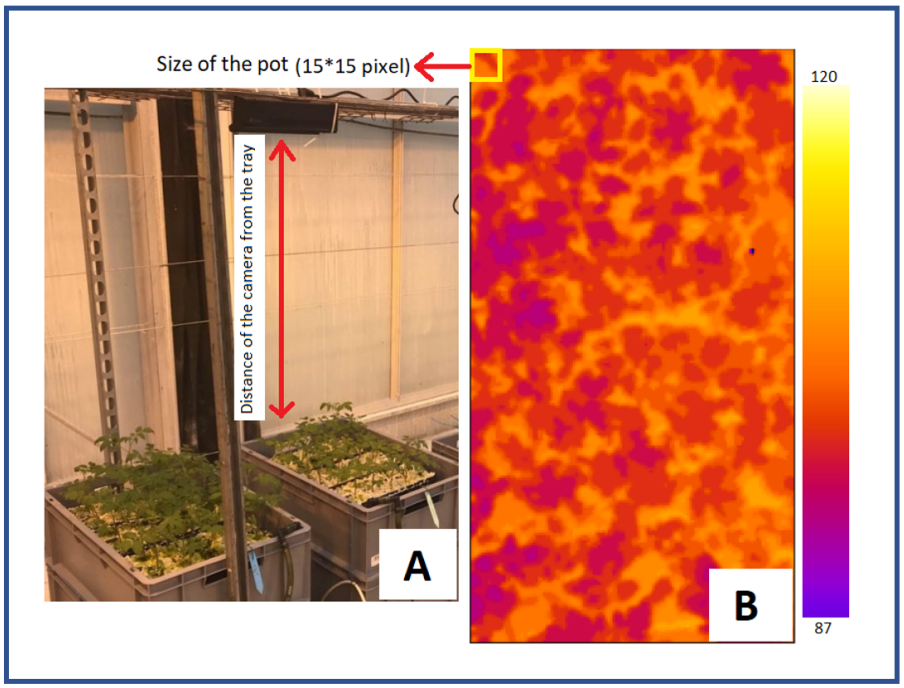


Figure 2.24 – Panel A, view of the image acquisition system. Panel B, colorized depth map with look-up Table « fire ». The levels are indicated in cm.

Qualitative signal analysis

Typical growth signals recorded with the imaging setup of the previous section are shown in Fig. 2.25. Different components are visible. First, a global linear trend shows the global growth of the plant, which gets closer and closer to the camera. Second, some oscillations are visible at the exact day period. These oscillations correspond to the so-called circadian rhythm that allows plants (like most living organisms) to synchronize their physiology with the daily period of light, maximizing their ability to benefit from sunlight and minimizing energy loss when the light is not available [76]. A third component is visible and corresponds to a higher frequency pattern that occurs when leaves are replicated and produce some mechanical movements. For illustration, we propose in Fig. 2.25, an example of the growth curve for control plants and plants under stress (hydric or salt stress).

Design of a Fourier feature space

Before the introduction of low-cost depth imaging operating in the infrared domain, the monitoring of plant growth was somehow limited to the average growth rate. The

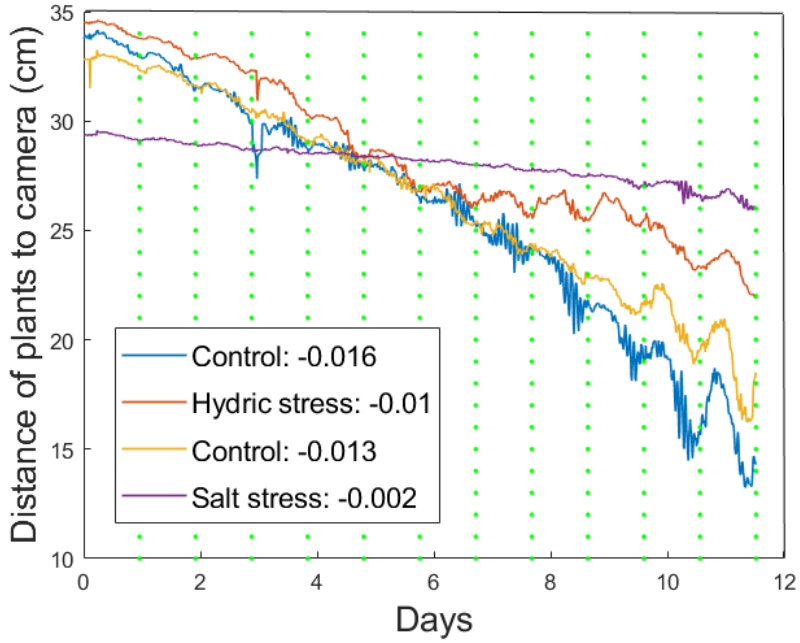


Figure 2.25 – Spatial average of the distance map $x(t)$ to the camera in cm as a function of time in various conditions. The values indicated in the inset correspond to average growth rates in centimeter per minute.

monitoring of growth, as shown in Fig. 2.25, allows quantifying this process in more detail. Following the qualitative description of the previous section, we propose to design a feature space based on a small set of numbers to encode the growth signal. The spatial average distance map to the camera $x(t)$ is first detrended with a daily linear trend which for the plants and growing duration selected in this study stands as a reasonable model. This produces the daily signal

$$y(t,n) = x(t) - (Gr(n) \times t + K) \tag{2.5}$$

with

$$t \in [nT, (n+1)T)]$$

and

$$(Gr(n), K) = \text{argmin}_{\tilde{Gr},\tilde{b}} \sum_{t=nT}^{t=(n+1)T} (x(t) - (\tilde{Gr} \times t + \tilde{K}))^2 \tag{2.6}$$

where $Gr(n)$ simply measures the daily growth rate on day $n = \{0, 1, 2, ..., 12\}$ of the canopy and T is the daily period. Then, since the cellular processes can, from a theoretical

biologic point of view [76], be assumed to be synchronized with the daily period of the sun, we decompose $y(t,n)$ as a Fourier series[77] and compute the modulus of its fundamental

$$c_1(n) = \sqrt{a_1(n)^2 + b_1(n)^2} \tag{2.7}$$

with

$$a_1(n) = \frac{2}{T} \times \int_{nT}^{(n+1)T} y(t) \cos\left(\frac{2\pi}{T}t\right) dt \ , \tag{2.8}$$

$$b_1(n) = \frac{2}{T} \times \int_{nT}^{(n+1)T} y(t) \sin\left(\frac{2\pi}{T}t\right) dt \ . \tag{2.9}$$

The daily period T is assumed constant over the two weeks of observation. Energy in the daily sinus of amplitude $c_1(n)$ is found to represent more than 95% of $y(t,n)$ over the two weeks of observation. Therefore $c_1(n)$ constitutes a good approximation of the amplitude of the circadian cycles [75, 76]. However, to also capture the presence of the high-frequency movements, we also consider the harmonic distortion rate

$$HDR(n) = 100 \times \sqrt{\frac{E(n) - \frac{1}{2} \times c_1(n)^2}{\frac{1}{2} \times c_1(n)^2}} \ , \tag{2.10}$$

where $E(n)$ is the energy of the detrended signal $y(t)$

$$E(n) = \frac{1}{T} \times \int_{nT}^{(n+1)T} y(t)^2 dt \ , \tag{2.11}$$

which captures the relative energy in the replication phenomenon of the leaves, which causes the high-frequency patterns. The instantaneous growth rate Gr obviously enables in Fig. 2.25 to differentiate between the control plant and a stressed plant. However, when representing growth in a (HDR, c_1) graph, as in Figs. 2.26 and 2.27, with time as a parameter, it appears that these trajectories clearly differ also between control and stressed plants. Also, all recorded trajectories start with a low amplitude of the fundamental, then approximately after 6 days, an increase of the harmonic distortion rate with diminution of the amplitude of fundamental follows, and after 10 days a decrease of the harmonic distortion rate and an increase of the fundamental. Trajectory learning could be undertaken in this feature space once we have more of these experiments. Here, we rather focus on the assessment of the added value of this extended feature space Gr, c_1, HDR when compared to the usual single scalar feature space based on the individual growth rate Gr. We propose a feature space which sums up the global shape of the temporal

trajectories of Figs. 2.26 and 2.27 and consider the following 5-dimensional feature vectors

$$feat = (Gr, \max(c_1), \min(c_1), \max(HDR), \min(HDR)) \ . \tag{2.12}$$

We propose to compare the added value of this feature space when compared to the classical single growth rate of Gr alone for two applications.

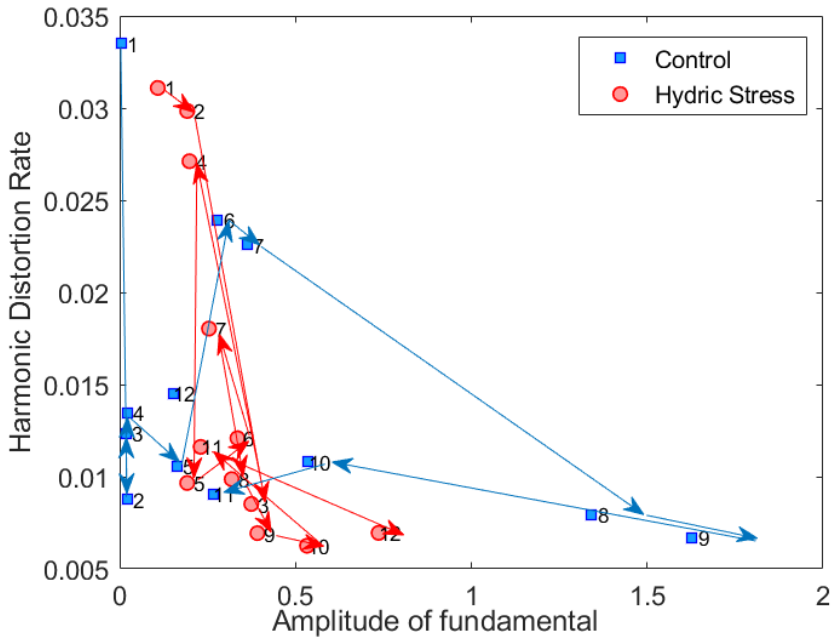


Figure 2.26 – Temporal trajectories of growth represented in a HDR, c_1 graph for control in blue and hydric stress in red. The arrows indicate the flow of time.

Applications and discussion

Best observation time: One of the biological questions that we can address with our feature space is how to discriminate the plants which are in control condition from plants under stress. When is the best time to observe the differences between the plants in different situations? To this purpose, we computed the Mean Square Error (MSE) of the feature vectors between control and stressed plants as a way of feature space contrast

$$MSE = \frac{1}{5}\sum_{i=1}^{5}(feat_c(i) - feat_s(i))^2, \tag{2.13}$$

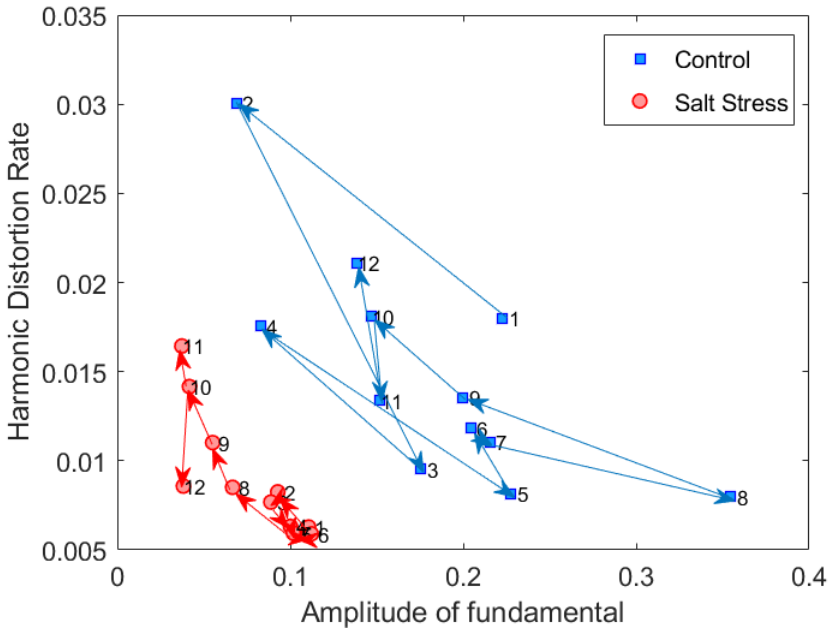


Figure 2.27 – Same as in Fig. 2.26 but with red for salt stress.

where $feat_c$ is the feature vector for the control and $feat_s$ for the stressed plant. As shown in Figs. 2.28 and 2.29, the extended feature space based on $feat$ of Eq. (2.12) can be above the basic reference of the growth rate at early stages, but it is difficult to have a definite point on this since only two records were done. However, it seems obvious from Figs. 2.28 and 2.29 and Fig. 2.25 that the contrast (MSE) between stressed plant and control plant is much higher after ten days with the extended feature space proposed here and the usual single growth rate. This is the best observation time if one wants to take benefit from the extended feature space based on Fourier analysis proposed here.

Stress detection: To further assess the interest of the proposed extended feature space of Eq. (2.12) we go beyond contrast metrics and implement a supervised detection scheme to classify stressed plants from control plants. The feature extracted from Eq. (2.12) are fed to a support vector machine (SVM) with linear kernel. The effectiveness of SVM classifier is evaluated by the K-fold cross-validation K=10. [78]. For comparison, the individual growth rate is computed and applied to the same SVM classifier. Small images of size 15 by 15 pixels are created in-depth maps, as shown in Fig. 2.24. This corresponds to the size of a single pot. The performance of the classification is given in

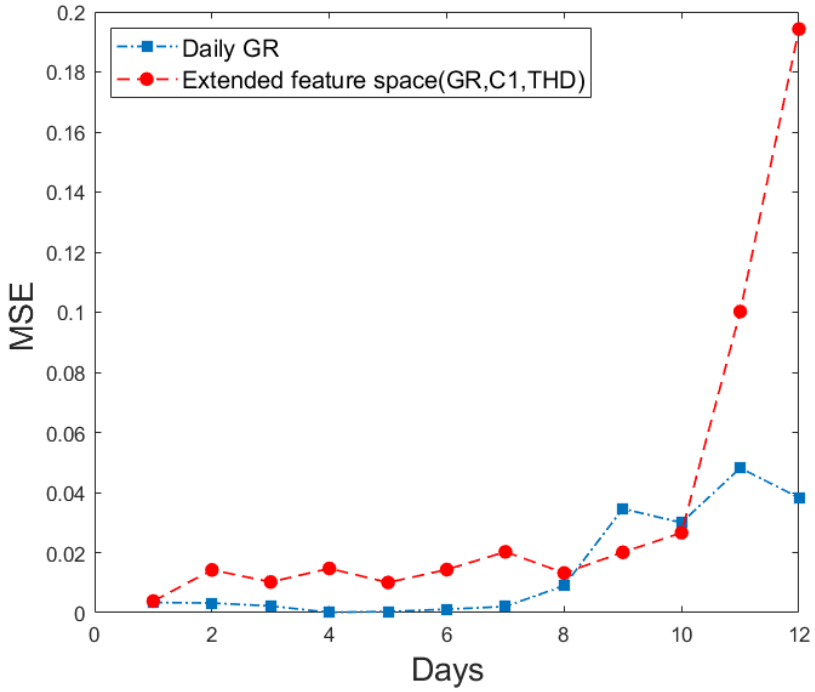


Figure 2.28 – Contrast between control and salt stress for the sole growth rate (Daily GR) and for the extended feature space of Eq. (2.12) computed by the MSE of Eq. (2.13).

terms of accuracy based on the following formula

$$accuracy = \frac{TP + TN}{Total} \ , \tag{2.14}$$

where TP stands for the true positive and TN for the true negative. The accuracies for classifications based on the extended feature space of Eq. (2.12) Moreover, the sole growth rates are given in Table 2.8. This clearly demonstrates proceeds between 4% and 9% of accuracy when the feature space is extended to the Fourier-based features of Eq. (2.12). Measuring the amplitude of the circadian cycle and the distortion rate of these circadian cycles improve efficiency to discriminate control from stressed plants.

Conclusion and perspective

We have applied, for the first time to the best of our knowledge, low-cost depth imaging to the monitoring of plants growth in canopy observation scale. This imaging solution enables to monitor plant during day and night at stages where they start to touch and

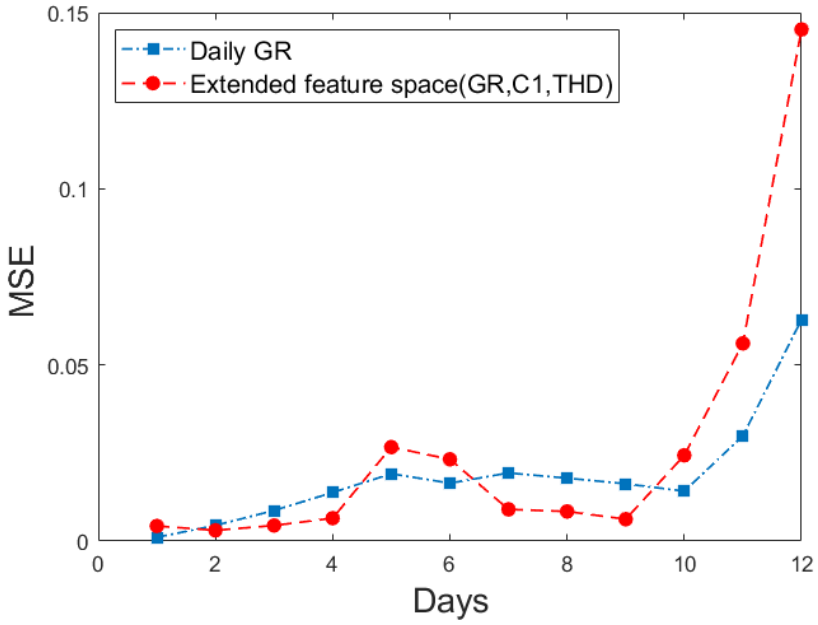


Figure 2.29 – Same as in Fig. 2.28 but with hydric stress.

	Accuracy K-fold K=10
GR FS Control, Hydric Stress	83.1%
GR FS Control, Salt Stress	94.8%
Extended FS Control, Hydric Stress	92.2%
Extended FS Control, Salt Stress	99%

Table 2.8 – Accuracy for the SVM K-fold (K=10) cross-validation classification between stressed plants from control plants with a feature space only based on growth rate (GR FS) or based on our extended feature space of Eq. (2.12) (Extended FS).

overlap each other. This approach constitutes an interesting alternative to the imaging system developed in the previous section which was limited to day observations and non touching plants. The novelty of this system also comes from the fact that we modify here the usual practice of biologists. Usually when population of plants have to be monitored, biologists randomize their positions in order to avoid bias due to spatial inhomogeneity in the trial. Here, we deal with much smaller populations (100 plants under each camera)

and can ensure homogeneous environmental conditions to these plants. This enable to avoid randomization and allow to group plants of the same kind under a unique camera.

From a methodological point of view we have demonstrated that the situation can be understood as a signal processing problem. We designed a feature space based on Fourier analysis and illustrated its interest on agronomical applications. Accumulated data will enable in the future more applications in the direction of deeper understanding of the temporal trajectory of growth in this feature space.

Nonetheless, the imaging system and associated signal processing procedure is ready for large scale and has been used in the framework of the ANR Labcom Match in partnership with our institute and the AREXHOR compagny `https://www.astredhor.fr/arexhor-pays-de-la-loire-45764.html`. This compagny is testing bio-stimulating products developed by the domain of agro-chemical industry. In the process of testing such products it is important to have fast screening on small population of plants. A replication of our system has been installed in the compagny and a training has been organized on how to acquire the data. An online version of the code replicating our method has been developed by a master student from our group `https://sithamfr.shinyapps.io/GrowthData/` and is now used as a demonstrator to establish new industrial contacts.

Chapter 3

Low-cost machine learning

We have focused on developing low-cost imaging techniques in the previous chapter. We now pay out effort on reducing the cost of machine learning algorithms. As stressed in the introduction chapter, machine learning algorithms when used in supervised ways require considerable amount of annotated data. This is especially true with deep learning algorithms which need even more data to be efficient. Also, these deep learning algorithms reach their high performance to the price of the use of high-speed computational devices operated in GPU. In this section we first review the different ways to reduce the price of image annotation. We then present our contributions to this field when applied to plant imaging. We finally investigate low computational resources to perform deep learning analysis.

3.1 Approaches for fast image annotation

Manual annotation of images is necessary to establish ground-truth in supervised machine learning. A typical order of magnitude of the number of instances (pixel, object, image) is between 1000 and 10000 for a supervised deep learning algorithm to converge. When operated by experts such a task requires much effort, making it a labor-intensive, time-consuming, and error-prone task. There are different strategies to accelerate the creation of such ground-truth which can be organized in human-assisted annotation and computer-assisted annotation. We propose a detailed panorama of these strategies in this section.

3.1.1 Human-assisted image annotation

One way to speed up image annotation is to parallelize the work with several annotators. Numerous tools exist and different criteria can serve for the choice of the best platforms for speeding up human-based image annotation. We have made an extensive

review of them presented in Table 3.1 structured along the following questions:

— Who annotate?
— What level of annotation?
— What level of confidentiality?
— How much is the cost/quality of the annotation?

Who annotate?

Annotation can be split and done online through platforms by assigning micro-tasks to the crowd or micro-games to volunteers. Depending on the complexity of the annotation task, sometimes, it should be done with the collaboration of a team of experts in the field. David G.Stork in 1999, introduced the concept of *e-citizens* in *Open Mind Initiative* as a worldwide effort to develop intelligent software [79]. *Crowdsourcing* coined in 2006 by Jeff Howe[80], as using human collective-intelligence on the Internet to collect ideas, solves complex cognitive problems, and builds high-quality repositories. Thus, crowdsourcing is a recent phenomenon which can developed under various forms.

Citizen science games are one form of *Games with a Purpose* (GWAPs) [81, 82, 83] applications. The purpose behind GWAPs is to conduct scientific research with volunteers instead of scientific experts. Volunteering in this context is an altruistic activity in the form of playing games where members of a community contribute in terms of time, resources, and services to annotate specific data without being paid financially [84]. The *ESP game* [85] was the first citizen science game to label images with crowdsourcing. It can be used to determine what objects are in the image, but cannot be used to determine the location of each object. It was the reason that the *Peekaboom* [86], a web-based game that can segment objects in images, was introduced. *FoldIt* [87] is another example of GWAPs with the form of puzzles in order to advance knowledge about protein structures. The goal of *Eyewire* [88] is "map the brain" through users and discover neural connections. Last but not least, *Zooniverse* [89] is a crowdsourcing platform for the deployment of citizen science projects with the versatility of crowdsourcing throughout various domains.

Other available commonly used citizen science game platforms for image and video annotation tasks include, Image Parsing [90], M-OntoMat-Annotator [91], Photostuff [92], Spatial Annotation [93], Flicker [94], IBM EVA [95], Name-It-Game [96], Galexy Zoo [97], Valleywatch [98], Ask'nSeek [99], Tag around [100], SeaFish [101], KissKiss-Ban [102], Tag4Fun [103], EteRNA [104], Planet Hunters [105], Malaria Diagnosis Game

[106], BioGames [107], MalariaSpot [108], Verbosity [109], ASAA [110], Manhattan Story Mashup [111], Phetch [112], Matchin [113], Waisda [114].

Micro-task based crowdsourcing platform was appeared in 2005 by introducing *LabelMe* [115, 116] a web-based annotation tool which provides a way of building large annotated datasets by relying on the collaborative effort of a large population of users. Later, in 2007 *dedicated annotation services* [90] were developed to create high volume quality annotated images and video frames but at a high price. Finally, in 2008, Alexander Sorokin and David Forsyth presented *Amazon Mechanical Turk* [117], which efficiently outsourced image annotation task. The micro-task based crowdsourcing platforms [118] break down a large project into Human Intelligence Task (HIT). These micro-tasks are then distributed among the workers who get paid a monetary reward for each completed task [119]. Micro-task platforms are well suited to perform research, as they grant on-demand access to large crowds for various types of problems. There are several survey studies which are related to efficient, and effective crowdsourcing frameworks for creating large-scale well-annotated datasets on different application domains, for example, in computer vision[120, 121, 122, 123, 124], health-care [125, 126, 121, 127], bioinformatics [128, 129, 130] and agriculture [131, 132, 133].

Many traditional crowdsourcing frameworks could not provide a comprehensive platform for sophisticated data. It yields inconsistent quality, and confidentiality is a significant concern in crowdsourcing [134]. On the other hand, end-to-end annotation platforms have multiple advantages. For example, outsourcing companies can handle a large volume of data. They can complete such tasks with higher productivity with in-house trained annotators while ensuring the quality of the annotation services and annotate images based on the customized needs [134, 135].

The other option for collaboration annotation is using in-house annotation platforms [136]. The purpose of the in-house annotation platform is to provide a framework in which experts in the field can collaborate to do annotation in a team, in a private and secure environment. Some platforms provide specific features like project management. There is a feature to split the project into micro-tasks. In contrast, the generic ones are the online platforms that speed up the annotation process in terms of accessibility and provide the annotation layers in the format adopted for training machines.

What level of annotation?

There are three different levels of image annotation in computer vision. Figure 3.1, illustrates these different levels such as the image level, object level, and pixel-level [90]. Image level and object level annotations consist of information meant to describe an entire image. The general category of an image, whether or not it contains a particular feature of interest, or represents a specific type of scene are examples of image-level annotations. Pixel-level annotations are used to mark-up particular regions of interest (ROI) within an image. These annotations are used to localize individual objects within an image and to segment out ROIs from the background. Most of the image annotation tools offer the possibility to make region-based annotations [96]. There are various methods to execute region-based annotations, such as drawing a bounding box around an object. The advantage of the bounding-box selection is that it is swift; however, the disadvantage is that the selection is inaccurate and often selects much more image data than necessary. The polygonal method offers the possibility to make a more detailed selection by drawing a polygon around the object. This method is fast and more precise than the bounding box. Nevertheless, since it uses straight lines, it is still challenging to make a very accurate selection. By freehand drawing tool, one can draw a free line around an object, which enables exact selections. The distinct disadvantage is the time consumed in this approach. Object tracking on videos, semantic segmentation, and instance segmentation is possible by the region-based annotations.

What level of confidentiality?

Data privacy is another critical criteria which should be considered. Some platforms present a dedicated on-premise cloud for providing the privacy of data. Some platform providers pledge that the data will be removed from their servers after the annotations. Some platforms operate locally and do not need to put data on the remote server. For some, the data are available publicly on cloud services.

How much is the cost of the annotation?

The human-assisted platforms are at a different cost. Some are free and open-source, some are image-based pricing or hour-based, some are volunteer-based, and for some, the cost is based on the annotation task, single or multiple objects. There are many available online/offline, standalone, or crowdsource platforms used to annotate data. In

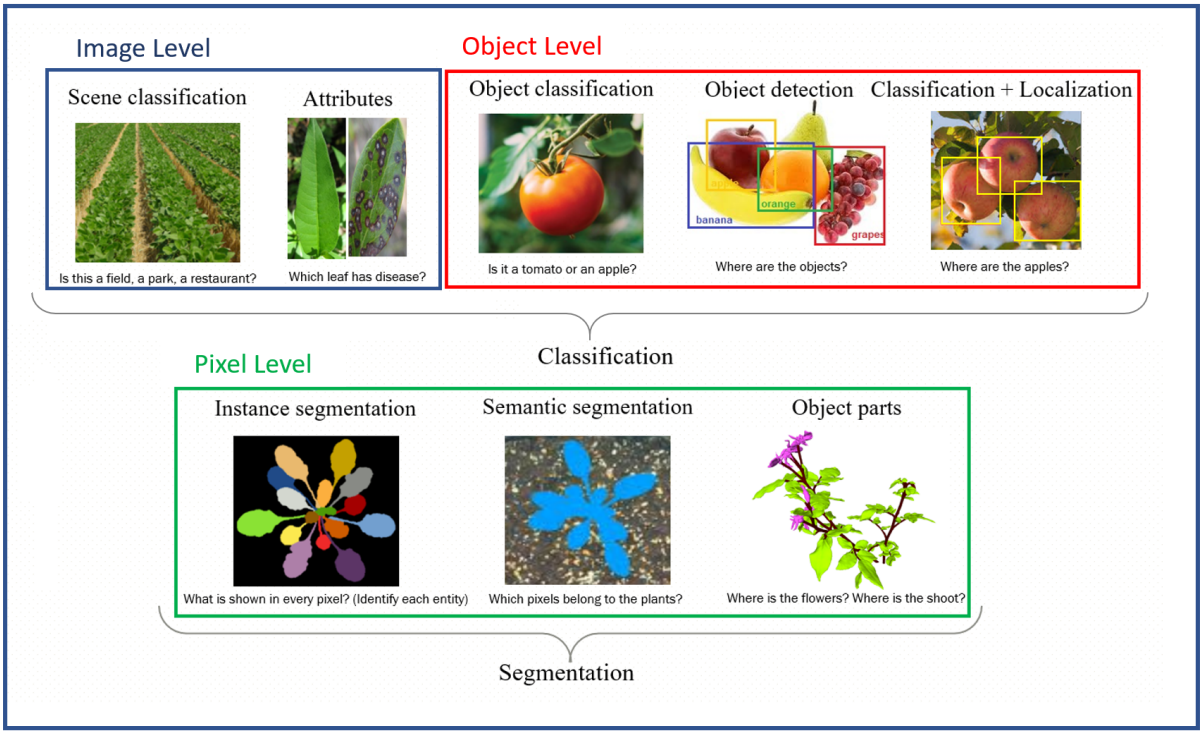


Figure 3.1 – Panel of computer vision tasks and associated type of annotation requested for supervised machine learning.

Table 3.1, we mentioned some of the well-known platforms widely used for image and video annotation tasks.

Platform	Collaboration					Level				Data Confidentiality		Cost		Assisted	
	Crowd			In-house											
	Game	Public	Privare	Generic	Specific	Image	Object	Pixel	Temporal (Video)	Public	Private	Free	Paid	ML assisted	Model assisted
Dataturks [117]	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-
clickworker [137]	-	✓	-	-	✓	✓	✓	✓	-	✓	✓	-	✓	-	-
Figure Eight (CrowdFlower) [131]	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-
Colabeler [138]	-	-	-	✓	-	✓	✓	✓	✓	-	✓	✓	-	-	✓
Phenotiki – Leaf Annotation Tool [6, 139]	-	-	-	-	✓	-	-	✓	-	-	✓	✓	-	-	-
samasource [140]	-	✓	✓	-	-	✓	✓	✓	✓	-	✓	-	✓	-	-
spare5 (Mighty AI) [141]	-	-	✓	✓	-	✓	✓	✓	-	-	✓	-	✓	-	-
CloudFactory [142]	-	✓	-	-	✓	✓	✓	✓	-	✓	-	-	✓	-	-
Infolks [143]	-	✓	-	-	-	✓	✓	✓	-	-	✓	-	✓	-	-
LabelMovie [144]	-	-	-	-	✓	-	-	-	✓	-	✓	✓	-	-	✓
Supervise.ly [145]	-	-	-	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	✓
VGG Image Annotator [146, 147]	-	-	-	✓	-	✓	✓	✓	✓	✓	-	✓	-	✓	-
Labelbox [148]	-	✓	-	-	✓	✓	✓	✓	✓	✓*	✓	✓	✓	✓	✓
Labelme [116]	-	-	-	✓	✓	✓	✓	-	-	✓	-	✓	-	-	
Labelme (Ketaro Wada) [149]	-	-	-	-	-	✓	✓	✓	-	-	✓	✓	-	-	✓
VATIC [150]	-	✓	-	✓	-	-	-	-	✓	✓	✓	✓	-	-	-
RectLabel [151]	-	-	-	✓	-	✓	✓	✓	-	-	✓	-	-	✓	-
ByLabel [152]	-	-	-	✓	-	✓	✓	✓	-	-	✓	✓	-	-	✓
FreeLabel [153]	-	-	✓	-	✓	-	-	✓	-	✓	✓	✓	-	-	✓
Prodigy [154]	-	-	-	✓	-	✓	✓	-	-	-	✓	-	✓	✓	-
TrainingData.io [155]	-	?	?	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
PixelAnnotationTool [156]	-	-	-	✓	-	-	-	✓	-	-	✓	✓	-	-	✓
COGITO [157]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	-	-
AISpotters [158]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	-	-
InfoSearch [159]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	-	-
HIVE [160]	-	-	✓	-	-	✓	✓	✓	✓	-	-	-	✓	✓	-
CVAT [161]	-	-	-	✓	-	✓	✓	✓	✓	?	-	✓	-	✓	-
BUNCH [162]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	✓	-
Scale.AI [163]	-	-	✓	-	-	✓	✓	✓	✓	-	✓	-	✓	?	-
OCLAVI [164]	-	✓	-	-	✓	✓	✓	✓	✓	?	✓	✓	✓	-	✓
microwork.io [165]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	?	-
Cytomine [136]	-	-	-	-	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓
SuRVoS [166]	-	-	-	-	✓	-	-	✓	-	-	✓	✓	-	-	✓
ilastik [167]	-	-	-	✓	-	✓	✓	✓	Volumetric data	-	✓	✓	-	✓	✓
FIAT [168]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	-	✓
TaQadam [169]	-	✓	-	-	-	✓	✓	✓	-	✓	✓	-	✓	✓	-
Humans in the loop [170]	-	✓	-	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
Playment [171]	-	✓	✓	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
WorkAround [172]	-	✓	-	-	-	✓	✓	✓	-	-	✓	-	✓	-	-
Diffgram [173]	-	✓	-	-	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
Linkedai [174]	-	✓	-	✓	-	✓	✓	✓	-	-	✓	-	✓	-	✓
haizaha [175]	-	✓	-	-	-	✓	✓	-	-	?	?	-	✓	-	-
ImageTagger [176]	-	-	-	-	✓	✓	✓	-	-	-	✓	✓	-	✓	-
VoTT [177]	-	-	-	-	✓	✓	✓	-	✓	✓	✓	✓	-	-	-
Anno-Mage [178]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	✓	-
CATMAID [179]	-	-	-	-	✓	✓	✓	✓	-	-	✓	✓	-	-	
EUDICO [180]	-	-	-	-	✓	-	-	-	✓	✓	✓	✓	-	-	-
Alp's Labeling Tool (ALT) [181]	-	-	-	✓	-	✓	✓	✓	-	-	✓	✓	-	✓	-
js-segment-annotator [182]	-	-	-	✓	-	-	-	✓	-	-	✓	✓	-	-	✓
LOST [183]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	✓	-
Quantius [180]	-	✓	-	-	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
SLOTH [184]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	-	✓
clay sciences [185]	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓
iSeg [186, 187]	-	-	-	✓	-	-	-	✓	✓	-	✓	✓	-	✓	-
Hasty [188]	-	-	-	✓	-	✓	✓	-	-	-	✓	-	-	✓	✓
ePAD [189]	-	-	-	-	✓	-	✓	✓	-	-	✓	✓	-	-	✓
COCO Annotator [190]	-	-	-	✓	-	✓	✓	✓	✓	-	✓	✓	-	✓	✓
SEASCAPE [191]	-	-	-	-	✓	-	-	✓	-	-	✓	✓	-	-	✓
handl.ai [192]	-	-	✓	-	✓	✓	✓	-	✓	✓	-	-	✓	-	-
UltimateLabeling [193]	-	-	-	✓	-	-	-	-	✓	-	✓	✓	-	-	✓
OpenLabeler [194]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	✓	-
Alturos Image Annotation [195]	-	-	✓	-	✓	✓	✓	-	-	✓	✓	✓	-	-	-
OpenLabeling [196]	-	-	-	✓	-	✓	✓	-	✓	-	✓	✓	-	✓	-
Yolo-mark [197]	-	-	-	✓	-	✓	✓	-	✓	-	✓	✓	-	✓	-
DeepLabel [198]	-	-	-	✓	-	✓	✓	-	✓	-	✓	✓	-	✓	-
Pixie [199]	-	-	-	✓	-	✓	✓	✓	✓	-	✓	✓	-	-	-
KNOSSOS [200]	-	-	-	✓	✓	-	-	✓	✓	✓	✓	✓	-	-	✓
pyImSegm [Segmentation2017, 201, 202]	-	-	-	✓	-	✓	✓	✓	-	-	✓	✓	-	-	✓
LC-Finder [203]	-	-	-	✓	-	✓	✓	-	-	-	✓	✓	-	✓	-
understand.ai [204]	-	✓	-	-	-	✓	✓	✓	✓	-	✓	-	✓	✓	-

Table 3.1 – List of commonly used crowd-source platform for image and video annotation tasks.

A last point to be discussed is the assessment of the quality of annotation when performed from crowdsourced annotation. The quality of crowdsourced annotations has been studied in different domains. The most common method for obtaining ground-truth annotations from crowdsourced labels is by applying a majority consensus heuristic. [117, 116]. Modeling annotation quality by applying confusion matrix [205] showed how repeated, and selective labeling increased the overall labeling quality on synthetic data [206]. Smyth et al. [207] integrated the opinions of many experts to determine a gold standard and later in[208] a method for combining prioritized lists obtained from different annotators were developed. Using annotator consistency to obtain ground-truth has also been used in the context of paired games and CAPTCHAs [85, 209]. Whitehill et al. [210] considered the complexity of the annotation task and the ability of the annotators. In [211], annotator models have been used to train classifiers with noisy labels. A system was proposed in [212] which actively asked for image labels that are the most informative and cost-effective. The reliabilities of online estimation of annotator is studied in [213].

3.1.2 Computer-assisted image annotation

We now review image annotation approaches where part of the work is done with assistance of the computer.

Machine learning guided platforms: To have an accurate computer vision system based on supervised machine learning, a lot of high-quality labeled data [214] is needed. Labelers must be extremely attentive as each mistake or inaccuracy negatively affects a dataset's quality and the overall performance of a predictive model. Thus, some platforms are assisted with image-processing and machine-learning algorithms to speed up the image annotation process. For instance, watershed marked [215], Deep Extreme Cut (DEXTR) [216], magic wand, and superpixel[217], are used in interactive learning platforms such as [167, 166] to provide fast, clean and accurate ground-truth data. Some others are using a pre-trained classifier or utilized active learning [218], MaskRCNN [219], for speeding up the human-assisted annotation.

Transfer learning: Transfer Learning [220, 221] is another interesting paradigm to bootstrap the learning phase of supervised machine learning where the network undergoes weight modification, which can be very time-consuming and may need an extensive set of images. Also, it prevents overfitting. Transfer learning starts with a network pre-trained on large datasets, such as ImageNet [222] or other specific available annotated datasets. Then, it uses those weights as the initial weights and fine-tuning the network

to the new task of interest. Typically, just the weights in convolutional layers are copied, rather than the entire network, including fully connected layers. This is very effective since many image datasets share low-level spatial characteristics that are better learned with big data. This strategy is widely used in machine learning-based plant phenotyping for accelerating the learning process. It has proven successful in many plant phenotyping applications such as disease diagnosis [223, 224], plant species classification [225] and root segmentation [226]. For the first time, we get the advantage of this strategy to transfer the weights learned in the RGB color domain to a different imaging modality such as chlorophyll fluorescence imaging for leaf segmentation application, explained in section 3.3.

Data augmentation: Another solution for providing adequate data with ground-truth for training supervised machine learning algorithms is the data augmentation strategy. Data augmentation encompasses a suite of techniques that obviates the need for manual image annotation and data limitation by enhancing the size and quality of training datasets [227]. The generated synthetic or simulated data by these techniques can help to probe data and add desire invariance, equivariance, and symmetry to the dataset. It can also reduce overfitting and regularize the solution space and improve the generalization of models since it increases the training set's diversity. Data augmentation can be performed by either image manipulation techniques or machine learning-based techniques. Image manipulation techniques transform existing annotated images available in the dataset by algorithms such as geometric (horizontal and vertical flipping, random scaling, random cropping, translation, rotation, shearing, stretching,...), photometric transformations (color and contrast jittering, sharpening, white balancing,...) mixing imaging and noise adding. Machine learning augmentations techniques such as feature space transformation, Gan-based data augmentation, and meta-learning create synthetic instances and add them to the training set.

Thre are two ways to add augmented data to the machine learning pipeline; the first one is the offline augmentation. This method is preferred for relatively smaller datasets and will increase the dataset's size by a factor equal to the number of transformations performed. The second option is known as online augmentation or augmentation on the fly. This method is preferred for larger datasets, that we cannot afford the explosive increase in size. Instead, we perform transformations on the mini-batches that will feed to the model. To have a robust invariant machine learning model to a variety of conditions such as translation, rotation, viewpoint, illumination, size, or a combination of them,

we need to study the dataset carefully. We can add diversity to or dataset by adding synthetically modified relevant data. This popular technique is widely used in plant phenotying applications [228, 229, 230, 228, 231]. We apply data augmentation by mapping Gaussian white noise by a model learn from small Arabidopsis fluorescence dataset to the same species RGB dataset to simulate the leave's texture structure and the thermal noise on the camera. In this way we increase the size of our fluorescence dataset by adding synthetic data in the study mentioned in section 3.3. Another approach to generate this diverse synthetic data can be from scratch programmatically by using simulators or game engines which is described in follow.

Simulation: The cost of data acquisition and annotation is high; using synthetic data can help to change this situation, and it is an important approach to overcome the problem of insufficient data and associated ground-truth in machine learning applications. This approach solves the data problem by either producing simulated data programmatically or using advanced data manipulation techniques to produce novel and diverse training examples. Synthetically generated datasets provide a reliable and cost-effective annotated data and guarantee a well-balanced dataset. Open source philosophy and access to reliable synthetic data generators and simulators can substantially boost simulated data development. Simulated data can be generated either when there is no equivalent available annotated data or meet specific needs or conditions that are not available in existing real data. For example, segmentation of roots in soil is an expensive and challenging problem and usually done in X-ray tomography. However, using purely synthetic soil and roots dataset and transfer learning approach, make it feasible with good results on simulated roots and on real roots even when the soil–root contrast is very low [226]. It is also possible to provide artificial images of plants using generative neural networks when large annotated plant image datasets for the purpose of training deep learning algorithms are lacking [232]. ElonSim as a simulator of seedling growth which incorporates parameters of the plant and parameters of the experimental imaging system acquiring the images are developed in [233]. This simulator opens the possibility to assess root segmentation algorithms. In this thesis, we generate synthetic data explained in section 3.2.1, to mimicking the existence of different kinds of weeds on the dense of plant mesh by developing a simulator.

3.2 Contributions to human-assisted image annotation

Vision reaction time to the visual stimulus was measured for various tasks [234], and it was demonstrated that the visual reaction time to visual stimulus is significantly faster compared to the auditory reaction time, especially when the goal is to discriminate the outliers. Eye-tracking devices can be used to record both field-of-view and gaze direction simultaneously, and it measures vital visual information such as fixations, gaze points, and saccades [235]. Therefore, one way to assist human in image annotation can consist in capturing the position of the eye a human expert while he analyze an image in order to offer a direct link between his eye and the computer. Such a strategy is accessible via the use of eye-tracking systems. There are two main types of eye-tracking devices such as screen-based, and glasses. We propose a contribution on each of these devices to speed-up image annotation.

3.2.1 Screen-based eye-tracking

Our first contribution to demonstrate the possibility to speed up image with screen-based eye-tracking is dedicated to the detection of weeds in dense plants from top-view.

We consider the situation of a culture crops of a high density of plants (mache salad) with the undesired presence of some weeds. Images were acquired with the imaging system fixed on a robot as displayed in Fig. 3.2. This plant science problem is important for field robotics where the mechanical extraction of weed is a current challenge to be addressed to avoid the use of phytochemical products. Acquisition trials, as visible in Fig. 3.2, were done under plastic tunnels without additional light. Some sample images are given in Fig. 3.3. Examples of weed detected in such images are shown in Fig. 3.4 to illustrate the variability of shapes among these wild types of weeds. The computer vision task considered in this study consists in detecting the weeds from the top view as shown in the ten real-world images of Fig. 3.3. This is challenging indeed since the intensity or color contrast between weed and crop is very weak. Also, due to the lighting conditions during acquisition, the global intensity may vary from one image to another. The contrast between weeds and plants rather stands in terms of texture since the shape of the plant considered is rather round while the weeds included in the dataset of Fig. 3.4 are much more indented.

Figure 3.2 – Global view of the imaging system fixed on a robot moving above mache salads of high density. RGB images are captured by a JAI manufactured camera of 20Mpixels with a spatial resolution of 5120x3840 pixels, mounted with a 35 mm objective. The typical distance of plants to camera is of 1 meter.

A ground-truth of the position of the weed in the ten images of Fig. 3.3 was produced under the form of finely segmented weed and bounding box patches including these weeds. The total number of weeds being relatively low (21), we decided to generate a larger dataset with synthetic images. To simulate images similar to the real images acquired, we created a simulator which places weeds (among the 21 found in real images) from the annotated weed dataset in images of plants originally free from any weed along the pipeline shown in Fig. 3.5. We generated a dataset of 150 synthetic images in which weeds were randomly positioned on high-dense plants.

Eye-tracker sampled eye positions of two observers during the execution of this task [236, 235]. The area of interest was recorded as rectangular patches. A patch is considered as including weeds if the average fixation time in this patch exceeds 1.04 seconds. The quality of visual annotation by eye-tracking is assessed in two ways. First, the visual annotation is directly compared with ground-truth, which shows an average 94.7% of all fixations on an image that fell within ground-truth bounding-boxes. Second, as shown in Fig. 3.6 eye-tracked annotated data is used as a training dataset in three machine learning

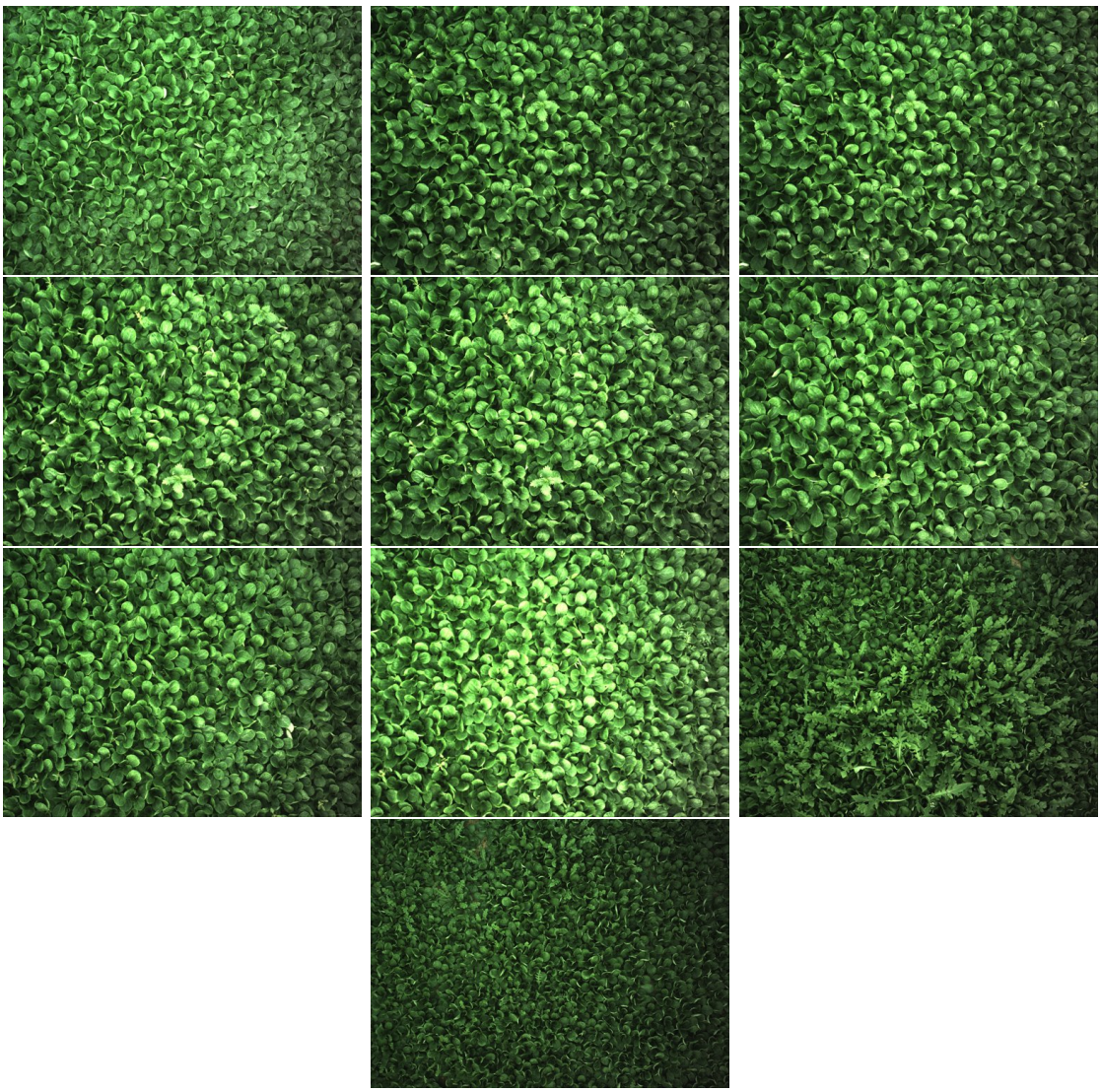


Figure 3.3 – Set of 10 RGB images from top-view for the detection of weed out of plant used as testing dataset in this study.

approaches and compare the recognition rate with the ground-truth. Handcrafted features adapted to texture characterization by three different approaches, including local binary pattern (LBP) [237], Haralick texture features [238], and Gabor filters[239], are extracted from the images. They are followed by a linear SVM [240] binary classifier. We assess the quality of the visual annotation by testing the trained classifiers by these three approaches that we shortly recall here.

Local binary pattern: Under the original form of [237] and as used in this study, for a pixel positioned at (x, y), local binary pattern (LBP) indicates a sequential set of

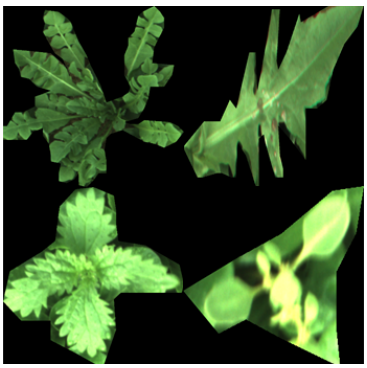


Figure 3.4 – Illustration of different types of weeds used for the experiment.

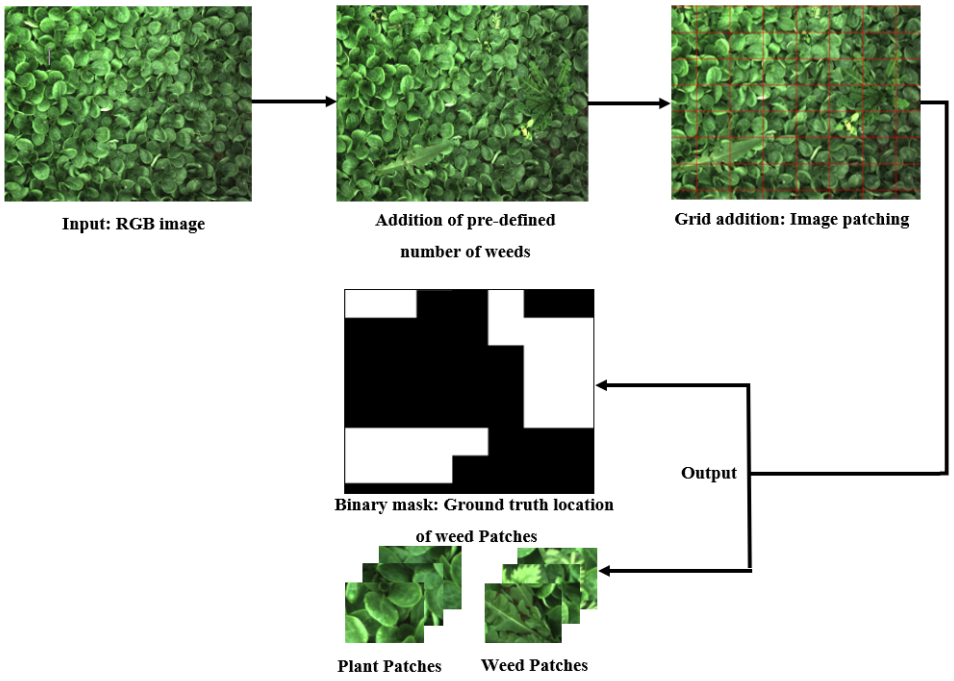


Figure 3.5 – Simulation pipeline for the creation of images of plant with weed of Fig. 3.4 similar to the one presented in Fig. 3.3.

the binary comparison of its value with the eight neighbors. In other words, the LBP value assigned to each neighbor is either 0 or 1, if its value is smaller or greater than the pixel placed at the center of the mask, respectively. The decimal form of the resulting

8-bit word representing the LBP code can be expressed as follows

$$LBP\,(x,y) = \sum_{n=0}^{7} 2^n s\,(i_n - i_{x,y}\,) \tag{3.1}$$

where $i_{x,y}$ corresponds to the grey value of the center pixel, and i_n denotes that of the n^{th} neighboring one. Besides, the function $\xi(x)$ is defined as follows

$$\xi\,(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0\,. \end{cases} \tag{3.2}$$

The LBP operator remains unaffected by any monotonic grey scale transformation which preserves the pixel intensity order in a local neighborhood. It is worth noticing that all the bits of the LBP code hold the same significance level, where two successive bits value may have different implications. The process of equation (3.1) is produced at the scale of the patch defined in the previous section. The $LBP(x,y)$ of each pixel inside this patch are concatenated to create a fingerprint of the local texture around the pixel at the center of the patch. Equations (3.1) and (3.2) are applied on all patches of an image.

Gray level co-occurrence matrix: A statistical approach that can well describe second-order statistics of a texture image is provided by the so-called gray level co-occurrence matrix (GLCM). GLCM was firstly introduced by Haralick et al. [241]. A GLCM is essentially a two-dimensional histogram in which the $(i,j)th$ element is the frequency of event i co-occurring with event j. A co-occurrence matrix is specified by the relative frequencies $C(i,j,d,\theta)$ in which two pixels, separated by a distance d, occurs in a direction specified by the angle θ, one with gray level i and the other with gray level j. A co-occurrence matrix is therefore a function of distance d, angle θ and greyscales i and j. In our study, as perceptible in images of Fig. 3.3, the weed-plant structures are isotropic meaning that they show no specific predominant orientations. As a logical consequence, and as already stated in similar weed classification problem using GLCM [242, 243, 244], choosing multiple orientations θ would not improve the classification performance. We therefore arbitrarily chose a fixed $\theta = 0$ which enables to probe on average leaves positioned in all directions. For distance, d, it is taken at $d = 2$ pixels which correspond to a displacement capable of probing the presence of edges, veins, and structures in the limb.

Gabor filter: It is a linear filter for texture analysis. Gabor Filters which are tuned to different frequencies and orientations are designed to localize different, roughly orthogonal, subsets of frequency and orientation information in the input image [239]. This

filters have been shown to possess localization properties in both spatial and frequency domain and thus are well suited for texture classification problems. In practice to analyze texture or obtain feature from an image, a bank of Gabor filter with number of different orientation are used. In this work, A bank of 4 Gabor filter oriented at an angle of $\theta = \{0, \pi/L, \ldots, \pi(L-1)/L\}$, where $L = 8$ were applied to the images to produce a feature space.

The feature space generated from each approach is used to train SVM binary classifier. Table 3.2 gives the average accuracy and standard deviation of the weed detector trained on perfect ground-truth and ground-truth computed from the eye-tracking records. Experimental results prove that visual eye-tracked annotated data are almost similar to in-silico ground-truth, and performances of supervised machine learning on eye-tracked annotated data are very close to the one obtained with ground-truth. Also, providing the annotated data by this approach is at least 30 times faster by comparison with manual annotation by the human on the same dataset.

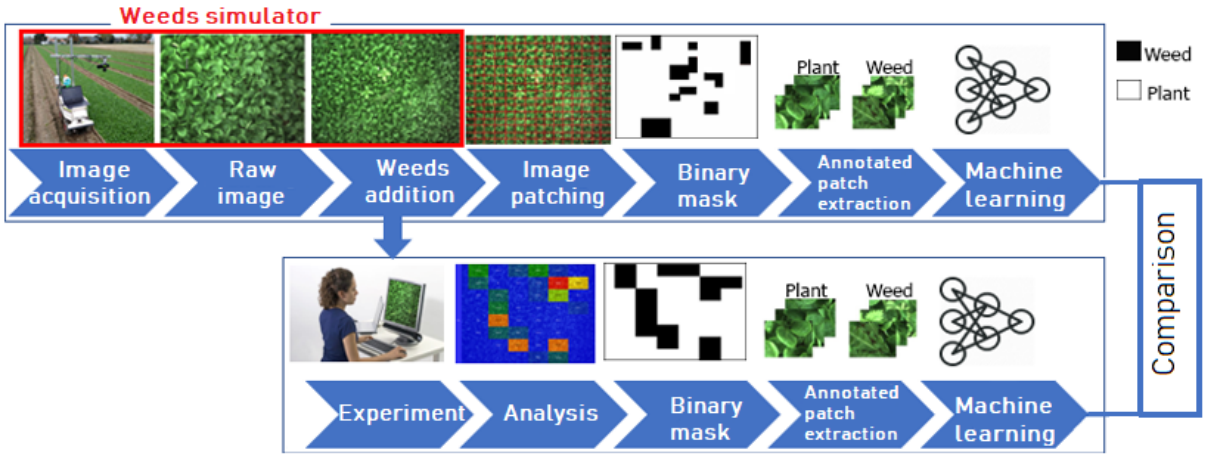


Figure 3.6 – General pipeline of comparison of eye-tracked annotated data with ground-truth.

Although the screen-based image annotation strategy accelerates the annotation time, it is a separate process after image acquisition to provide ground-truth. In the following section, we assess the value of various egocentric vision approaches to perform joint acquisition and automatic image annotation rather than the conventional two-step process of acquisition followed by manual annotation or using the screen-based eye-tracking device.

Methods	Recognition Eye-tracking First observer	Recognition Eye-tracking Second observer	Recognition Ground-truth
Local binary pattern	70.1 std: 0.69	71.33 std:0.27	73.1 std: 0.11
Haralick coefficients	65.7 std: 0.75	66.19 std:0.59	67.41 std: 0.22
Gabor wavelet filters	64.7 std: 0.41	62.37 std:0.77	67.2 std: 0.54

Table 3.2 – Classification performance for different annotated dataset of in-silico ground-truth and eye-tracked annotated data.

3.2.2 Egocentric head-mounted eye-tracking

We have demonstrated the possible interest of eye-tracking systems to speed up image annotation in the previous section with a screen-based device. We now investigate the value of an egocentric head-mounted device to speed up annotation. The term wearable "egocentric vision device" is used to designate all wearable imaging systems that record images from the first-person perspective. Images captured from egocentric devices are possible of high value since their field of view benefits from the attention of the person who wears the device and who is in charge of the targeted task to be done on the images. Reducing the field of view to a part of specific interest may reduce the inspected scene's complexity and help the automatic processing of the acquired images. This is expected to be especially useful in complex scenes, such as those found outdoor in agriculture and phenotyping in the fields. Also, some egocentric devices, namely head-mounted eye-trackers, can include capturing the ocular position of the annotator during the recording of the videos. This would, in theory, open the possibility to annotate images directly while acquisition and annotation are usually two separate steps. Such use of egocentric devices opens the possibility to conduct these steps jointly and hence reduce annotation time. However, eye-trackers can never be perfectly calibrated, and their practical value in terms of performance and time is still to be assessed to speed up annotation which is what we propose here.

For the first application of egocentric devices to accelerate annotation, we consider, as a proof of concept, a standard problem in computer vision for plant phenotyping. We choose the detection, i.e., segmentation, counting, and localization of apples in color images.

This task has been addressed in many ways, including recently, with deep learning. This canonical problem is challenging for computer vision since it includes self-occlusion of multiple instances, occlusion by the shoot of the apple trees, the variation of illumination, clutter from the self-similar background, variety in sizes and colors of fruits, and many more. Also, this computer vision problem is significant to be solved for various agricultural applications such as the design of automatic harvesting, automatic estimation of the fruit pack out, variety testing, and many more. A visual abstract of the proposed joint image acquisition-annotation process is illustrated with apple detection in Fig. 3.7.

Egocentric (first-person) vision is a relatively new research topic in the field of computer vision which is increasingly attracting the interest of understanding human activities [245, 246, 247, 248], object detection [249, 250], creation of models of the environment with different levels of precision [251, 252], perception of social activity [253], user-machine interactions [254], driving assistance [255], or medical applications [256, 257, 258], etc. There are different types of egocentric systems, such as smart glasses, action cameras, and eye-trackers. Based on the processing capabilities, embedded sensors, like the one used in this study, are now more and more used in conjunction with egocentric video analysis [254]. Features such as hand appearance, head motion give essential cues about the attention, behavior, and goals of the viewer [259, 260, 261, 262]. In our case, we also used the fact that usually in egocentric vision, salient objects of interest tend to occur at the center of the image since they attract the viewer's attention [263, 249].

In this work, we primarily used an eye-tracking system to perform an egocentric vision to speed up image annotation. The use of eye-tracker to speed up image annotation has been proven useful for annotation with a screen-based system in [264, 265, 266]. These researchers demonstrated a possible gain of time for annotating 30 (approximately) by comparison with manual annotation. Here, we use, for the first time to the best of our knowledge, an embedded eye-tracking system under the form of glasses (see Fig. 3.7) to jointly conduct image acquisition and annotation and thus extend the result of [264, 265, 266].

Object detection in agricultural conditions has been investigated with a large panel of computer vision approaches [267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279]. In the early works, like in [267], methods were handcrafted both from the hardware side and the software side. Nowadays, it is more common practice to use standard RGB cameras, and base the detection of apples on supervised machine learning methods learned end-to-end via deep learning like in [278, 279]. Such modern methods, neural network-

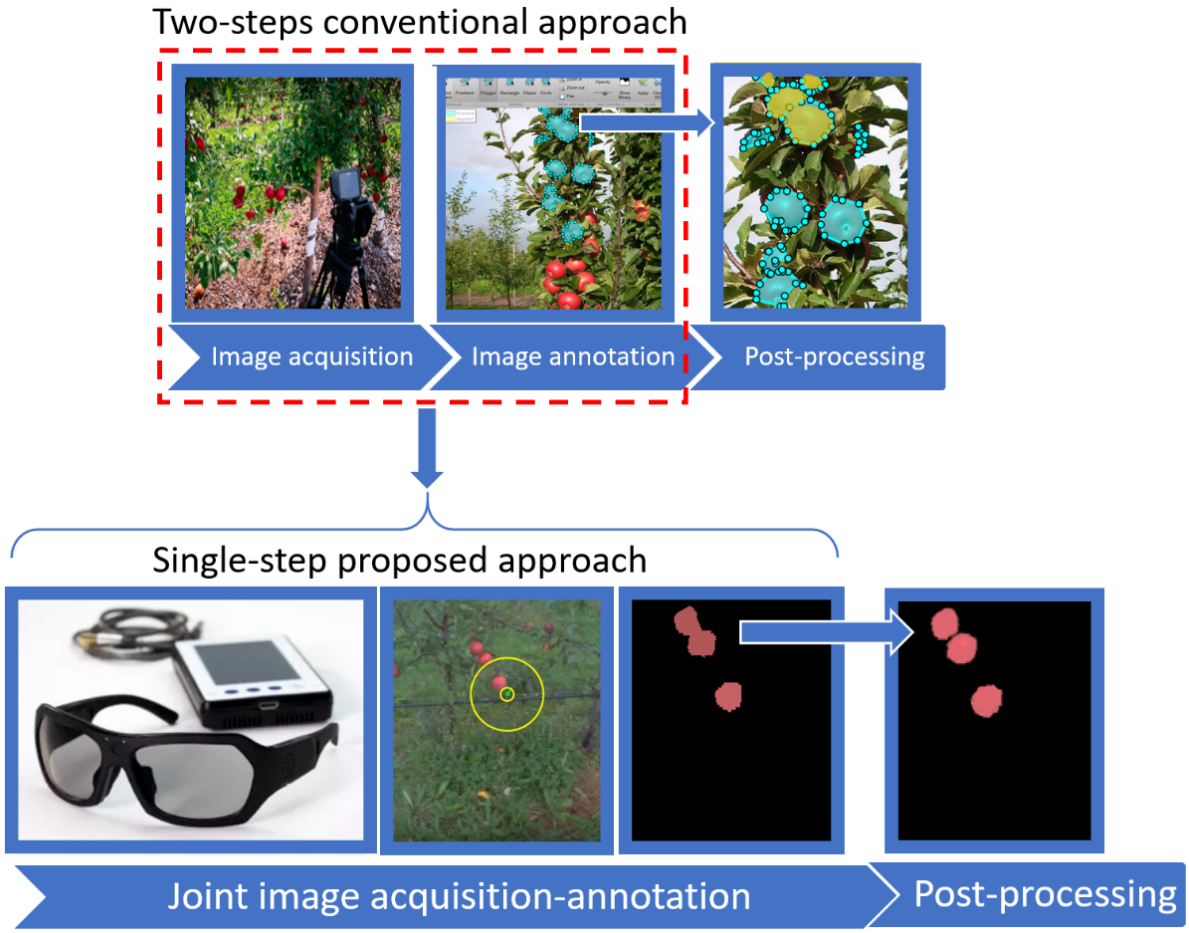


Figure 3.7 – Visual abstract of the egocentric head-mounted eye-tracking study. Red dotted-line is the conventional two steps of the acquisition and annotation process. We jointly perform image acquisition and image annotation by the use of a head-mounted egocentric device, which simultaneously captures images and the gaze of the person who wears the device and takes benefit of these to annotate images automatically. It is to be noted that the post-processing step to separate touching annotated objects is not included here. It remains a step necessary in the conventional two-steps approach and our proposed single-step approach.

based, show high performances but require a large amount of annotated images. Manual pixel-wise annotation is, in general, a time-consuming operation, taking approximately 1.5 hours per 100 images (308×202 pixels). In practice, apple detection is also challenging because of illumination conditions [280, 281, 282]. In this study, we will not provide a novel method to detect apples automatically. Instead, we will investigate the possibility to perform acquisition and annotation of apples in an orchard environment simultaneously

by using head-mounted egocentric devices. Indeed, while there has been significant recent interest in fruit detection, segmentation, and counting in orchard environments, the cost of providing a unified annotated dataset of the fruit on trees makes it the bottleneck in the state-of-the-art literature [283].

Egocentric vision device

The egocentric imaging system used was a VPS-16 head-mounted eye-tracking glasses equipped with stereoscopic cameras in the nose bridge, a front camera with a diagonal coverage of 88 degrees, and an audio microphone sampling at 10 kHz. The front camera was calibrated with the eye-tracker before the acquisition. The visual task defined to the wearer was to find apples on the targeted trees. The acquisition time was nearly 90 seconds for the whole dataset (calibration time included). This acquisition time is quite similar to the time required with a digital camera fixed on a tripod or handheld. It would need to be located in different positions to cover all apples located on a tree. The distance of the viewer and the tree was set approximately to one and a half meters. The viewer counted the number of apples as evidence of the ground-truth, which was recorded via the audio microphone. Fixation points were recorded by the eye-tracker to investigate how they can serve to annotate apples on the trees automatically.

Dataset

With the sensor described in the previous subsection, we generated a new dataset of 10 videos (25fps) from 10 various apple trees in the orchard environment captured by egocentric head-mounted glasses eye-tracker. The total number of extracted images from the entire dataset was 24618 frames.

A fundamental parameter of eye-tracking analysis depends on the definition of the fixation and the algorithm used to separate fixation from saccades [284]. Fixation refers to a person's point-of-gaze as they look at a stationary target in a visual field. Although the mean duration of a single fixation may depend on the nature of the task [285], numerous studies have been done to measure the average duration for a single fixation [286, 287, 288, 285, 289, 290, 291, 292, 293, 294]. The mean fixation duration for visual search is 275 msec, and for tasks that require hand-eye coordination, such as typing, the mean fixation can be 400 msec [285]. Among our dataset, the number of frames received at least 275 msec was 419 frames. On two days at midday, the acquisition was made on different weather conditions at the orchard of INRAE Angers, France. No difference was

found in the results of the data coming from the two days. This dataset includes a variety of apple colors together with apple and foliage density, which are representative of the dataset found in the literature for apple detection [295, 296, 297]. Due to the complexity of each orchard tree, the illumination and environment itself, different natural colors were found in the images, including various shades of green, red, yellow, brown, or gray for foliage grass, apples, and tree trunk.

A ground-truth was created by manual annotation of the raw color images in approximately 54 seconds per image by using the Image Segmenter application in MatLab 2017a. A sample of raw color images from different apple trees and their corresponding manual ground-truth is illustrated in Fig. 3.8. For the whole dataset, which consists of 419 images, it roughly took 6 hours to annotate all images manually. These manual annotations were generated for evaluation of the accuracy of the egocentric vision methods presented in the next section.

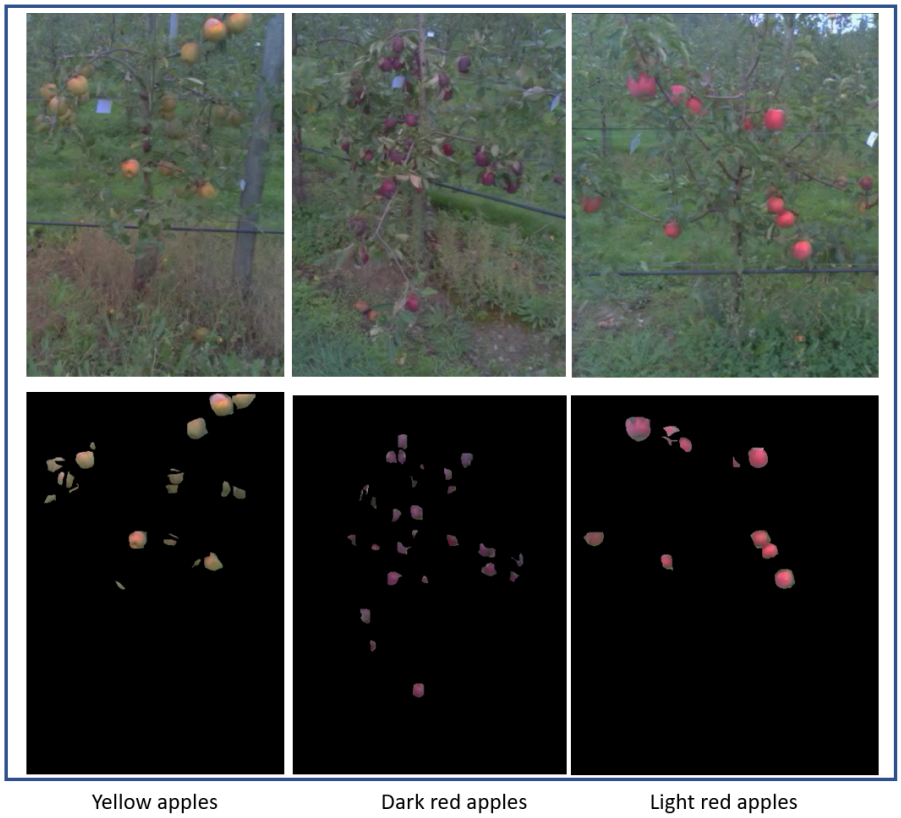


Figure 3.8 – Example of RGB images of apple trees from our dataset and corresponding ground-truth manually annotated.

Image processing pipeline

In this section, we present the image processing pipeline developed to automatically annotate apples from the attention areas captured with egocentric vision. A global view of this pipeline is depicted in Fig. 3.9 and includes three main steps: Image pre-processing, segmentation, and performance evaluation.

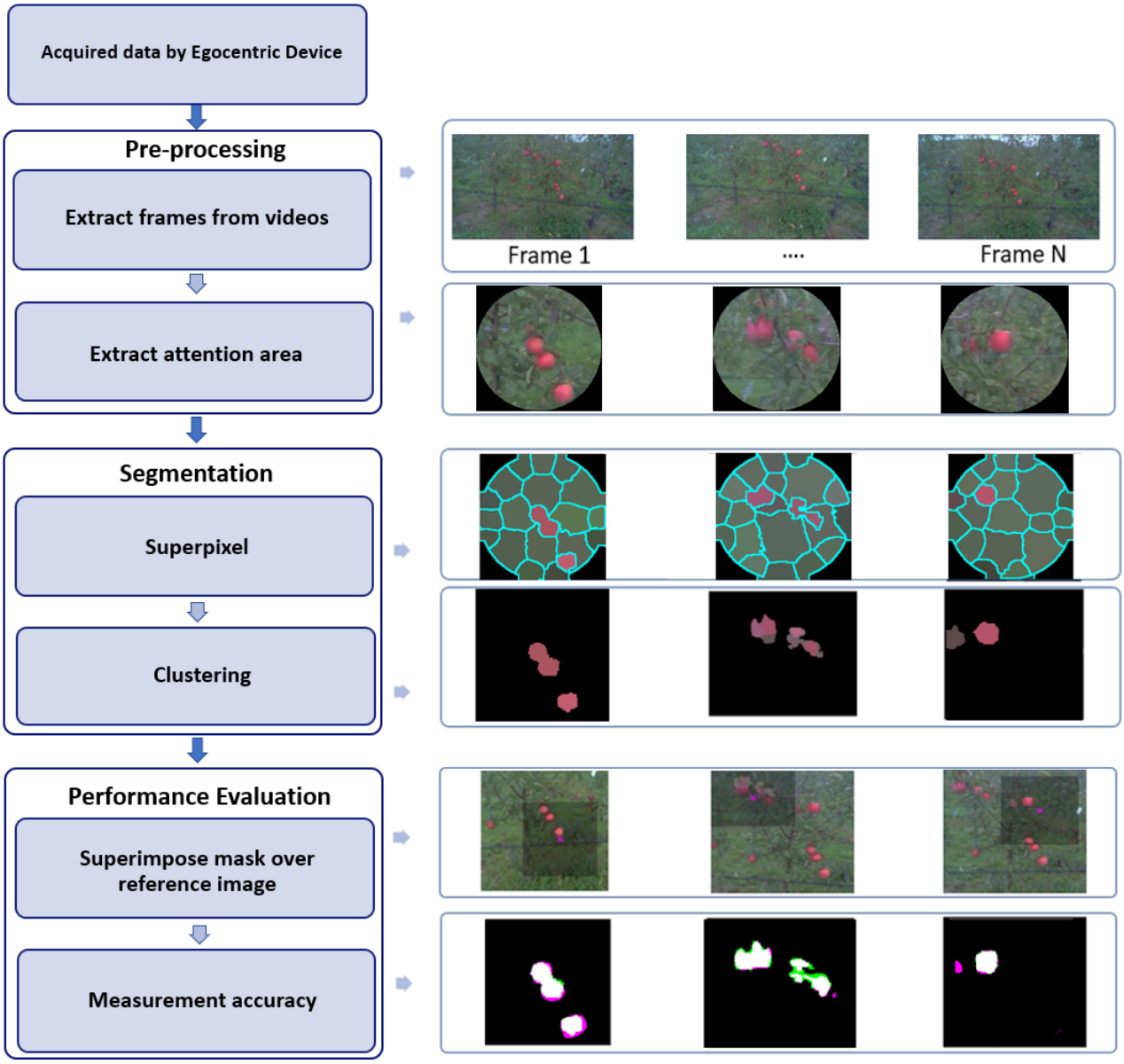


Figure 3.9 – Three steps image processing pipeline proposed to automatically segment apples from attention area captured with egocentric devices.

The pre-processing started with the extraction of the frames with a resolution of 960×544 pixels from recorded videos. Next, an attention area was extracted from each frame based on egocentric priors. The extraction of this attention area constitutes the main contribution of scientific research. Several strategies have been tested and are presented in the next section.

The pre-processed images were then segmented with a standard approach for apple detection similar to the one presented in [283, 298, 299, 300, 301]. A classical superpixel technique (SLIC) [217] was applied, followed by a simple non-supervised clustering technique chosen as K-means [302] to select superpixels corresponding to apples. To keep the size of superpixel independent of the size of the attention area we defined the number of superpixels as the ratio of

$$N = \frac{A}{S}, \tag{3.3}$$

where A represents the size of the attention area, and S the size of an average apple, which is equal to 900 pixels in our dataset.

To simplify the images, the tree-labels (blue in our case) and sky parts were removed by applying color thresholding (optimized on a small dataset) in the RGB color domain on the superpixel segmented attention areas, as shown in Fig. 3.10. The number of cluster K was found optimal for $K = 2$ and was applied to feature space composed of (R, G, B, H, S) respectively for Red, Green, Brightness, Hue, and Saturation from each superpixel. The cluster with the smaller size was considered as the apple cluster based on the assumption that the background occupied the largest area in the attention area. Because the blue part was withdrawn and that no green apple was present, this optimal value of $K = 2$ is reasonable for our use-case of apple detection in the orchard. Indeed, the local complexity in attention areas extracted from the egocentric devices is limited to objects on a background with a contrast of color. For other use cases, where local contrast between object and background could depend on other features (size, texture, shape, etc.), it would be necessary to adapt this segmentation.

Finally, the segmented apples were superimposed over the original image for qualitative assessment, localization and compared with the manual binary ground-truth to compute the segmentation accuracy via the Dice $Dc(X, Y)$ and Jaccard index $J(X, Y)$ given by

$$Dc(X, Y) = \frac{2 * |X \cap Y|}{|X| + |Y|}, \tag{3.4}$$

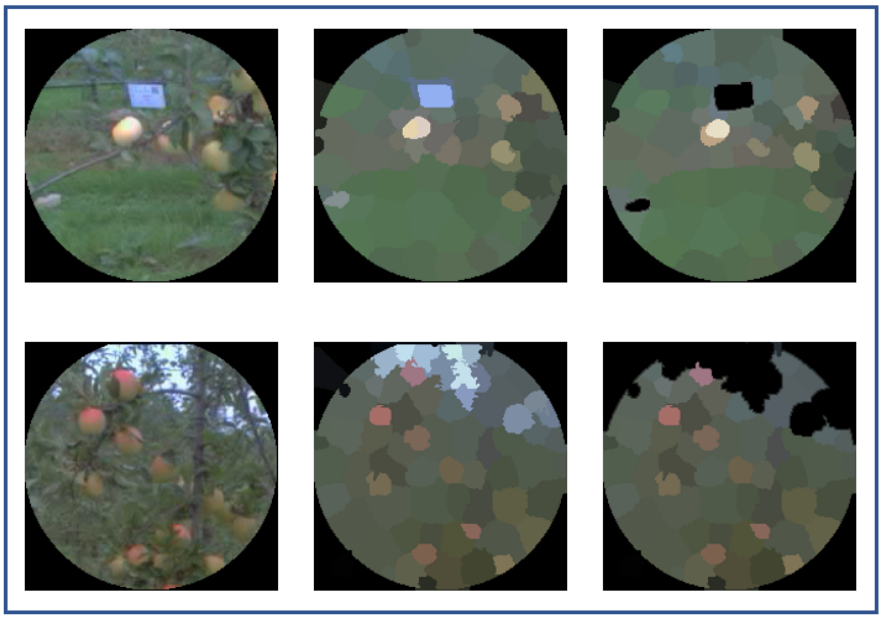


Figure 3.10 – Color thresholding to remove blueish color belonging to the sky or blue tree-labels on superpixel segmented attention areas. Each row represents from left to right: the attention area, superpixel segmented attention area, and the thresholded one, respectively.

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}, \tag{3.5}$$

where X and Y represent the segmented image and the ground-truth respectively.

In addition to the segmentation of apples, counting and localization were also computed in the following way. For object counting, we counted the number of connected components among detected objects which shared sufficient overlap with ground-truth. An empirical threshold of 75 percent was chosen for the overlap. The probability of good detection was computed as

$$PD = \frac{TP}{TP + FN}\ , \tag{3.6}$$

with TP number of true-positive objects and FN number of false-negative objects. We also computed the probability of true-negative rate as

$$TNR = \frac{TN}{TN + FP}\ , \tag{3.7}$$

with TN number of true-negative objects and FP number of false-positive objects. In localization, the Euclidean distance between the centroïd x_i of detected objects X_i and the

centroïd y_j of objects Y_i with a maximum intersection with ground-truth was computed as

$$d(x_i, y_j) = \sqrt{(u_{x_i} - v_{y_j})^2 + (u_{x_i} - v_{y_j})^2} \ , \tag{3.8}$$

with u and v which stands for Cartesian coordinates in the images and

$$j = \arg\max_{j_0} |X_i \cap Y_{j_0}| \ . \tag{3.9}$$

The average distance

$$d = \frac{1}{N} \sum_{i=1}^{N} d(x_i, y_j), \tag{3.10}$$

was computed over all detected objects sharing sufficient overlap with ground-truth. Here again, a threshold of 75 percent of overlap was chosen. Distance d represents the average shift error of localization of apples with an egocentric device from manual ground-truth.

Attention area from eye-tracking

In this section, we present strategies that we developed to extract attention areas from the eye-tracking devices to perform joint acquisition-annotation after passing these areas to the image processing pipeline of the previous section.

Selection by eye-tracking glasses

The first approach extracted attention areas via the viewer fixation computed from the egocentric eye-tracking glasses. In order to fix a threshold, a gazing position was recorded when the same fixation position was observed during an interval of 6 frames, as calculated by

$$fi = Fps * fd \ , \tag{3.11}$$

where fi is the frames interval, $Fps = 25$ is the number of frames per second, and fd is the average fixation duration, which was set as 275 msec.

Despite careful calibration before the acquisition, small shift errors of alignment between the front camera of the device, and the gazing point of the viewer can occur. Therefore, we extended the attention area around each gazing position with a given radius to compensate for the remaining small shift error of calibration of the eye-tracker. An illustration of the creation of an attention area around a fixation point is provided in Fig. 3.11. A systematic analysis of the evolution of the average segmentation accuracy as

a function of the radius of the attention area around each gazing position was undertaken.

It is shown in Fig. 3.12 and demonstrates a non-monotonic evolution culminating at a value corresponding to triple the size of an average apple size in our dataset. Consistently this optimal value was found to be very close to the maximum shift error of calibration of the eye-tracker found in the whole dataset. For too small attention areas, due to the shift error, apples can be missed. For too large attention areas, the segmentation process fails to detect all apples correctly in the area due to the complexity of the scene.

Figure 3.11 – Construction of attention areas. (a) The average diameter of an average apple is 30 pixels in our dataset; (b) Cross indicates the center of the gaze of the annotator. There is a shift error from the apple of (a). The maximum distance of the gazing point with the center of the closest object is found at 169 pixels ; (c) Chosen attention area with a size of 180 × 180 (pixels).

Selection by screen-based eye-tracking

For comparison with the attention area created with the egocentric eye-tracker directly acquired in the orchard, we also generated an attention map from the gazing point recorded with a screen-based eye-tracker. Of course, this approach is less interesting for the gain of time than the previous one with the head-mounted eye-tracker since it does not allow a joint acquisition annotation. However, screen-based eye-tracker is more accurate than head-mounted ones and thus are expected to constitute a reference serving as an upper bound in terms of quality of annotation with egocentric vision. The experiment was performed on a screen with a resolution of 1920 × 1080 (pixels) while the eye movements of the viewer were recorded with an SMI binocular remote eye-tracker [303].

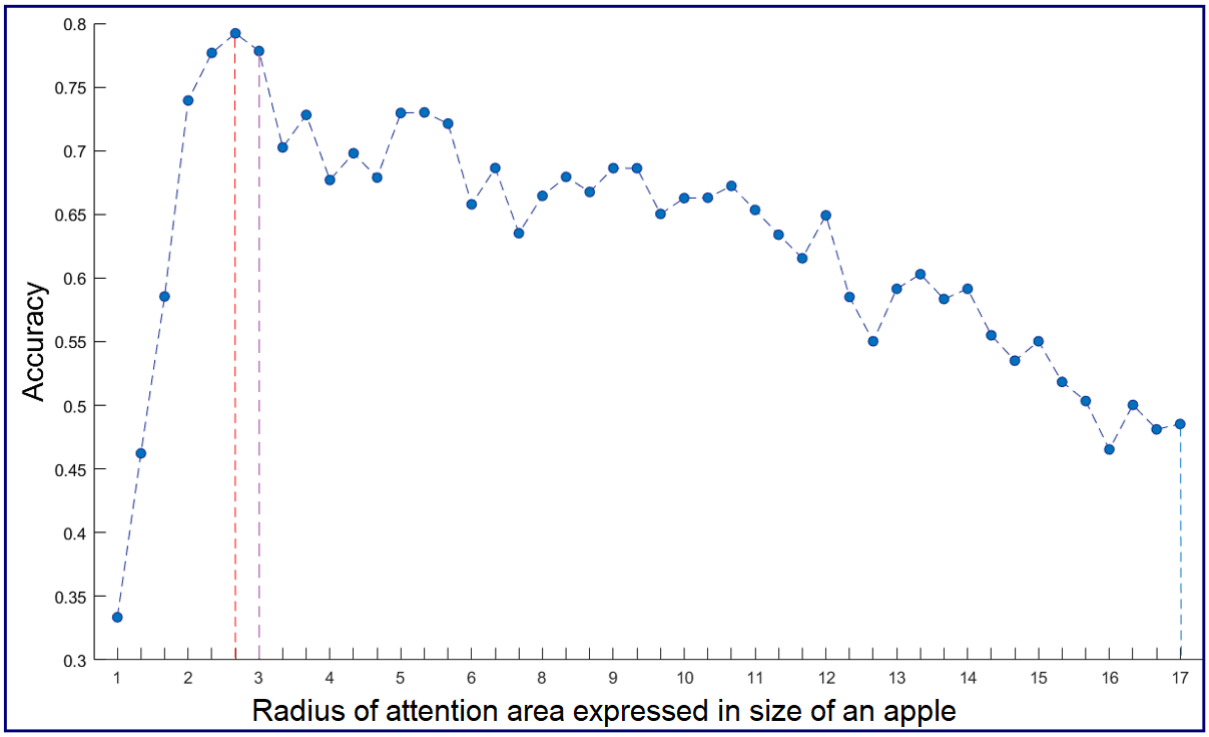


Figure 3.12 – Apple segmentation accuracy as a function of the radius of attention area expressed in the size of apples taken as 30 pixels. Maximum accuracy achieved when the radius size of the attention map is equal to 80 (160×160 pixels) corresponding to the red dotted line. The purple dotted line corresponds to the maximum gaze shift error of (169 pixels) between eye-tracker and ground-truth when computed on the whole dataset.

In this approach, for each apple tree, we peaked out one frame, which included all the apples.

The annotation protocol was the same as the previous method. Each image was displayed to the viewer, who was asked to find the apples on the trees. The locations of the fixations of the viewer were recorded at 60 Hz. For a fair comparison, the attention area diameter around each recorded fixation was taken at the optimal value found for the eye-tracking systems embedded in glasses.

A comparison of the accuracy of the screen-based eye-tracking recording and the recording with eye-tracking embedded in glasses was conducted. Figure 3.13 shows under the form of heatmap visualization of the attention of the viewer. The precision and accuracy of the produced gaze points with the screen-based eye-tracker were found higher than when using the head-mounted eye-tracker. The average shift error of Eq. (3.10) was found 125 pixels less with the screen-based eye-tracker than with head-mounted eye-tracker.

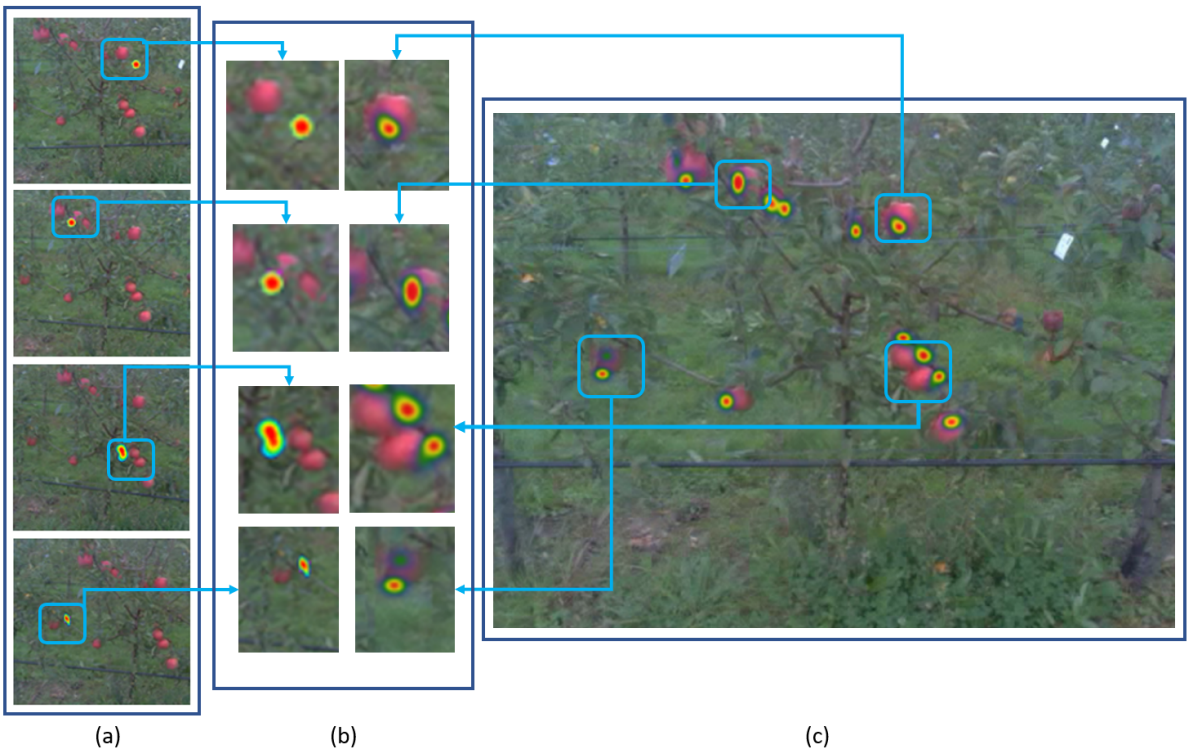


Figure 3.13 – Heatmap visualization of the attention of the viewer captured by head-mounted (glasses) eye-tracker (a) versus the screen-based eye-tracker (c). (b) Comparison of heatmap generated by the glasses eye-tracker (left) vs. heatmap generated by screen-based eye-tracker (right).

Attention area without eye-tracking

Other strategies were developed to extract attention areas for comparison with performances obtained with eye-tracking systems.

Full-frame

In this approach, the attention map was considered as the full-frame recorded by the camera. Thus, in Fig. 3.9, instead of a small patch of the entire original image, the full original image was directly transmitted to the superpixel segmentation. Such a choice assumes that the camera field of view is already a focus of the overall field of interest for the human annotator in charge of detecting apples.

Egocentric Prior

In this approach, we assumed, as often done in egocentric vision [249], that the viewer's attention is focused at the center of the frame. Therefore, we selected the attention area, as a disk at the center of the image with the size of 180×180 (pixels), for a fair comparison, with the other developed approaches eye-trackers.

Saliency Map

As the last method to compute an attention area, we turned toward a computational approach in charge of numerically identifying areas of interest. Such a concept has been developed in the computer vision literature under the name of the saliency map. Saliency acts as a local filter that enhances regions of the image which are standing out relative to their adjacent parts either in terms of orientation, grey level, and color contrast [304]. Introduced in [305], saliency is inspired by the mechanisms of human visual attention and the fixation behavior of the observer.

There are numerous computational models for salient object detection. In this study, for illustration and without any claim of optimality, we used the algorithm proposed by [306], which computes saliency map in images using low-level features and is proposed with codes included for reproducible science. Saliency maps were thresholded to binary masks following the fixed threshold procedure described in [306]. Each connected component of the binary saliency map served to produce an attention area. For a fair comparison with the other approaches, attention areas with a size of 180×180 (pixels) were chosen.

Results and discussion

We are now ready to compare the result of the different approaches proposed for apple detection by extracting attention areas through an egocentric vision from the perspective of a joint acquisition-annotation process.

As shown in Table 3.3, the best average performances (highlighted in bold) in terms of segmentation accuracy of apples are obtained with the eye-tracking-based methods. Challenging images and resulting annotation with eye-tracking-based methods are provided in Fig. 3.14 for qualitative assessment. Overall, the screen-based eye-tracker provides the best result but only slightly above the one obtained from the glasses eye-tracker. This embedded glasses eye-tracker, despite its substantial shift errors, is highly valued since it enables a joint image acquisition and annotation. The saliency approach provides a result

close to the one obtained with the baseline method (Full Frame). This could certainly be improved with a systematic benchmark of other saliency methods of the literature. However, a fundamental reason for the failure of the saliency approach, which would be common to all generic saliency maps, is that saliency is, so to say, attracted by contrasted objects which may not be apples (for example stems, leaves, items in the background, data matrix positioned in the field to identify trees). As a consequence, saliency creates much true-negative in attention areas since the task of detecting apples does not specifically drive it. In contrast, human attention focuses on the apple as captured by eye-tracking systems.

Interestingly these results are consistent for the three tasks, segmentation, counting, and localization assessed. This demonstrates the robustness of the interest of eye-tracker devices for annotation. Eye-tracking systems, such as the two different types used in this study, can be considered as expensive devices (typically between 10 to 20k euros currently). It is interesting to see that the egocentric prior approach gave the third-best performance, and this could be accessible with any camera embedded on glasses (for 10 to 100 euros).

Method	Dice	Jaccard	Good detection	True-negative rate	Shift error	Time (second)	Time Gain
Full Frame	0.24 ±0.22	0.21 ±0.16	0.31 ±0.20	0.17 ±0.72	174.11 ±34	880	24
Glasses eye-tracker	**0.78** ±0.08	**0.64** ±0.08	**0.84** ±0.16	**0.09** ±0.07	**15.97** ±11	1960	11
Screen-based eye-tracker	**0.85** ±0.09	**0.77** ±0.13	**0.88** ±0.12	**0.09** ±0.13	**2.37** ±1.86	3240	6
Egocentric Prior	0.46 ±0.36	0.38 ±0.31	0.54 ±0.39	0.28 ±0.23	84.82 ±7.25	1960	11
Saliency	0.27 ±0.13	0.16 ±0.08	0.42 ±0.45	0.51 ±0.17	7.21 ±8.28	2358	9

Table 3.3 – Performance of apple detection with the five approaches developed for extraction of attention area in the pipeline of Fig. 3.9. Each column corresponds to an average over the 10 trees of the dataset. Dice and Jaccard assess in percentage the quality of segmentation via Eq. (3.4) and (3.5) , good prediction and true-negative rate assess in percentage the quality of object detection via Eq. (3.6) and (3.7) and shift error of Eq. (3.10) assesses in pixels the quality of good localization. The time corresponds to the approximate annotation time for the whole dataset in seconds. Time Gain indicates the ratio of manual annotation time over automatic annotation time obtained with the egocentric devices. Time was measured on a windows machine with an Intel Xeon CPU and 32.0 GB RAM by MatLab 2017a.

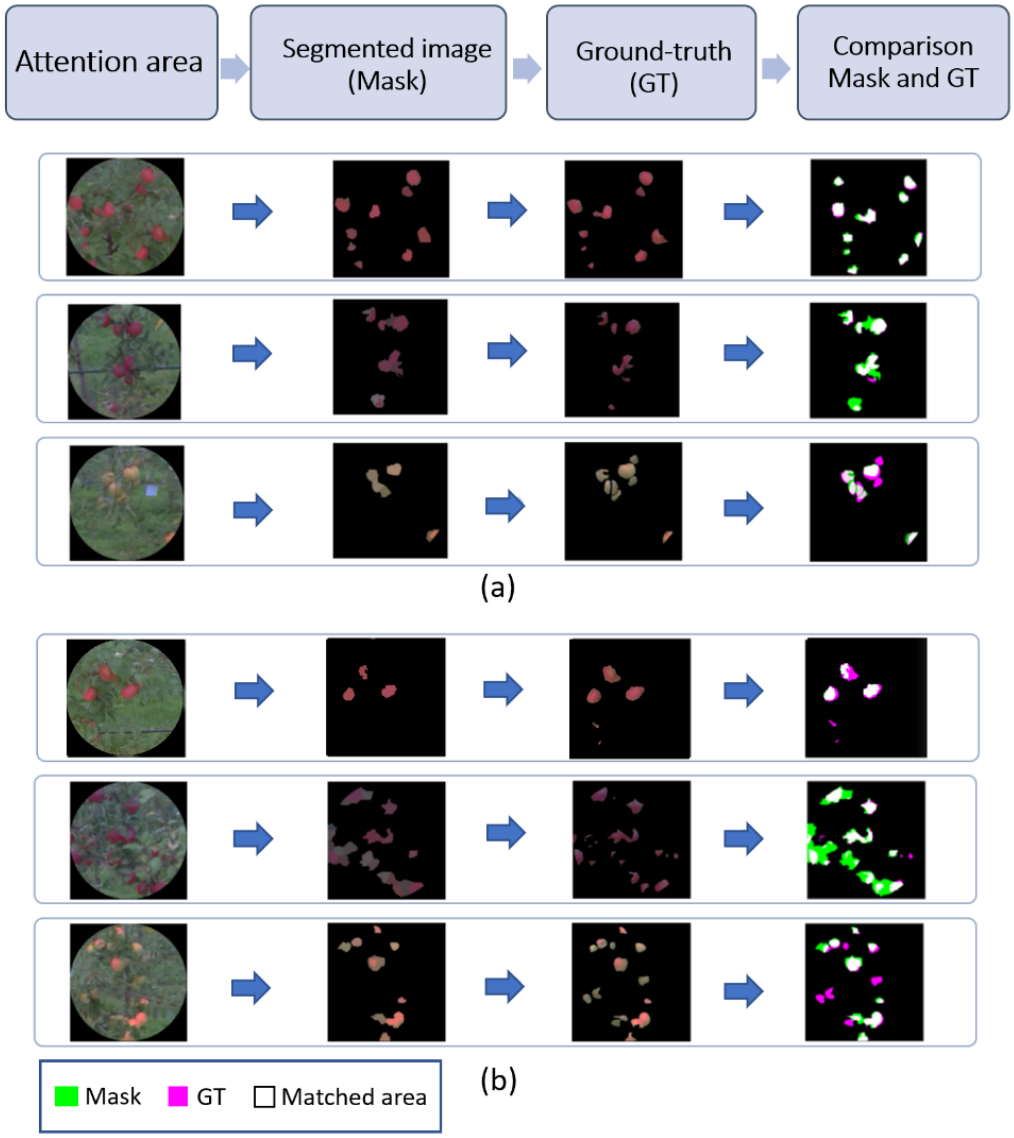


Figure 3.14 – Qualitative assessment of results. From left to right, an example of the attention area captured by eye-tracking, automatic annotation obtained from the proposed image processing pipeline of Fig. 3.9, ground-truth manually recorded, and comparison of manual ground-truth and automatic segmentation. (a) provides examples of good performance; (b) shows some challenging conditions where more errors are found (missed detection, false detection).

The value of the obtained results concerning segmentation, counting, and localization is to be assessed in terms of timing. As expressed in section (3.1), acquisition time with an egocentric device is comparable with acquisition time with any standard camera. Therefore gain of time is to be compared on the annotation time only. This timing is provided

in the last column of Table 3.3 for automatic annotation based on the image processing pipeline applied to extracted attention areas. Without any surprise, the Full Frame approach, which requires no computation of attention map, is the fastest method. The second most rapid methods are the egocentric prior and glasses eye-tracker. The screen-based eye-tracker method, which gave the best performance in terms of apple detection, comes with the slowest timing. However, these timings for automated annotation are to be compared with the timing requested to a human annotator to annotate all apples in the dataset manually. The estimated timing is 6 hours for the 419 frames. The gain of time for all methods is presented in Table 3.3. Saliency, as presented here, could be criticized since many other variants of the saliency map could be tested and possibly provide better results. In terms of timing, however, we believe the performances are realistic, and it was worth mentioning them here.

All in all, the glasses eye-tracker method appears as a good trade-off between speed and annotation performance. For this head-mounted device, the gain in performance is about 11, which is smaller than what was found in the literature with desktop eye-trackers for object detection [264, 265, 266]. This difference may come from the fact that in this literature, the task targeted were relatively more straightforward and required less post-processing. Optimization of the code could thus increase the gain in time. We are currently investigating all these perspectives.

Conclusion

We have assessed the value of egocentric imaging devices to perform acquisition and automatic image annotation jointly. This was illustrated with apple detection in orchards, which is a challenging task for computer vision applied to phenotype or agriculture. Despite shift errors in the calibration of egocentric imaging devices, the performance of the detection of apples from the gazed recorded areas was found to be very close to the one obtained from the manual annotation. The compensation for these shift errors was obtained by applying a standard non-supervised segmentation algorithm only in attention areas centered on the gazing positions captured by the egocentric devices. Specific interest was shown for head-mounted eye-tracking systems with an estimated gain of time compared to manual annotation found 11 times faster with a non-GPU-accelerated software.

This first use of egocentric vision to speed up image annotation opens interesting perspectives, especially in plant phenotyping. The task here was focused on apple, but the approach is indeed generic. Thus, it would be interesting to extend the applicability to

other phenotyping items of interest. To remain on apple, this could include the determination of flowering stages or the detection of diseases. Additional technological services from egocentric vision could be tested to speed up annotation. For instance, this includes a sound recording, which could be coupled to automatic speech recognition for later fusion with information extracted from the captured images.

The pilot study presented here is promising. For a tool to be used by technicians and engineers in the field, it would be necessary to implement an ergonomic version of the software to experiment on an extensive network of users. The method was developed to accelerate image annotation with egocentric devices. Validation of the quality of the annotation was performed at various levels, including location, object detection, and pixel-wise segmentation. Another stage of validation of the quality of the annotation would be to train a machine learning algorithm on the annotated images and compare the performance with the manually annotated data.

3.3 Contribution to computer-assisted image annotation

In this section, we present our contribution to the computer-assisted image annotation. We investigate the possibility of data augmentation and transfer learning via synthetic images for the segmentation of leaves of seedlings from top view.

Due to heavy occlusion, variability in terms of size and shape, leaf segmentation is a challenging task from the computer vision perspective. One strategy to simplify the segmentation is to reduce the biological variability and focus on a limited amount of specific interest plant species. This has been undertaken in the CVPPP challenge since 2014 with a focus on few species, including *Arabidopsis Thaliana*, which serves as a reference for several fundamental biological questions. The effort to provide annotated data has enabled the significant improvement of state-of-the-art on segmentation performance. An open question is now how to transfer this knowledge obtained from RGB images on annotated plants either to other species or other modality of imaging. In this work, we focus on the translation of the knowledge gained from annotated leaves of *Arabidopsis Thaliana* in RGB to images of the same plant in chlorophyll fluorescence imaging.

Segmentation of *Arabidopsis* leaves in RGB images has been highly studied since the introduction of the CVPPP challenge. In 2014 and 2015, the contribution to this challenge proposed segmentation methods based on models [307, 308, 309], most of the following

participants have tackled the challenge with deep neural network [310, 311] . In the following, we will not propose any innovation on this side. However, instead of working on a standard architecture, we apply it for the first time on another imaging modality.

Chlorophyll fluorescence analysis is a non-destructive technique that has been developed to probe plant physiology [312]. Among all the chlorophyll fluorescence parameters that can be estimated, the maximum quantum yield of photosystem II (PSII) photochemistry ($Fv/Fm = (Fm - F0)/Fm$) is an indicator of plant stress [313]. Fluorescence chlorophyll by image analysis on the whole plant has been widely studied [314, 315, 316]. So far, to the best of our knowledge, analysis on individual leaves has yet not be tackled in top view images of *Arabidopsis Thaliana*.

Image simulation to boost machine learning received increasing interest in plant imaging [228, 232, 317, 318]. This can include standard data augmentation, sophisticated infography, or generative models from the convolutional network. In this communication, we generate the images from one imaging modality to learn on another imaging modality. This topic has been demonstrated possible, for instance, for life science applications in the medical domain [319] in cross-modal image synthesis and microscopy in a super-resolution problem [320]. We consider for the first time data augmentation from the synthesis of images from RGB imaging modality to chlorophyll fluorescence imaging in plant sciences.

Datasets

Three datasets coined *CVPPP*, *CSIRO* and *Real Fluo* are considered in this study. They are described in the following.

CVPPP : We use the dataset provided in the leaf segmentation challenge held as part of the Computer Vision Problems in Plant Phenotyping *CVPPP* workshop [321]. *CVPPP* dataset consists in 27 RGB images of tobacco plants and 783 RGB images of *Arabidopsis* wild and mutant plants. We considered only the *Arabidopsis* dataset in this study. All images were hand-labeled to obtain ground-truth masks for each leaf in the scene. These masks are image files encoded in PNG, where each segmented leaf is identified with a unique integer value, starting from 1, where 0 is the background.

CSIRO: To extend *CVPPP* dataset we use also generated synthetic images of top down view renders of *Arabidopsis* [322, 311]. The *CSIRO* dataset contains 10000 synthetic images (width x height: 550 x 550 pixels). Similar to *CVPPP* dataset, each RGB image has a corresponding leaf instance segmentation annotation: each leaf in an image is uniquely identified by a single color value, starting from 1, where 0 is background. All images are

stored in PNG format.

Real Fluo: For model testing, we use 38 real gray-scale fluorescent images of *Arabidopsis*. The PSI Open FluorCam FC 800-O (PSI, Brno, Czech Republic) was used to capture chlorophyll fluorescence images and to estimate the maximum quantum yield of PSII (Fv/Fm) on wild type control of *Arabidopsis Thaliana*. The system sensor is a CCD camera with a pixel resolution of 512 by 512 and a 12-bit dynamic. The system includes 4 LED panels divided into two pairs. One pair provides an orange actinic light with a wavelength of around 618 nm, with an intensity varying from 200 to 400 mol/m2/s. It provides a 2s pulse that allows the measurement of the initial fluorescent state (F0). The other pair provides a saturating pulse during 1s in blue wavelength, typically 455 nm, with an intensity of up to 3000 mol/m2/s. The saturating pulse allows the collecting of the maximum fluorescence (Fm). Fluorescence chlorophyll imaging was used in a dark-adapted mode after a dark period of 45 min [28] to produce maps with the fluorescent quantum efficiency $Fv/Fm = (Fm - F0)/Fm$. All these 38 images were manually annotated using the Phenotiki image analysis software [6, 139].

Segmentation by U-Net architecture

The segmentation of the leaves is considered to be a pixel-wise classification where the pixel of the contour of the leaves should be extracted from the rest of the images. Leaf contours allow separating leaves and thereby perform leaf segmentation with the help of a watershed transform. Each pixel is classified among three mutually exclusive classes: mask without contours, leaf contours, and background, which means, a three-component one-hot vector label every pixel.

We use U-Net model [323] for the pixel-wise classification. As shown in Figure 3.15, U-Net architecture is separated into three parts: the contracting/downsampling path, bottleneck, the expanding/upsampling path. The encoder-decoder type architecture with skipped connections allows combining low-level feature maps with higher-level ones, and enable precise pixel classification. A large number of feature channels in the upsampling part allows propagating context information to higher resolution layers. The output of the model is a three-channel label that indicates every pixel class, as shown in Figure 3.16. All activation functions in the convolutional layers are **R**ectified **L**inear **U**nits (ReLU) [324]. The last layer before the prediction is a softmax activation with three classes. Images and labels from all datasets were resized to width x height: 128 x 128 pixels. Using ground-truth, we created labels for the three classes, as shown in Figure 3.16. To

reinforce the contour class's learning, which is highly unbalanced, we replaced the encoder with a ResNet152 backbone pre-trained on ImageNet [325]. The decoder was not changed [323]. The resulting network has a total of 1,942,275 trainable parameters.

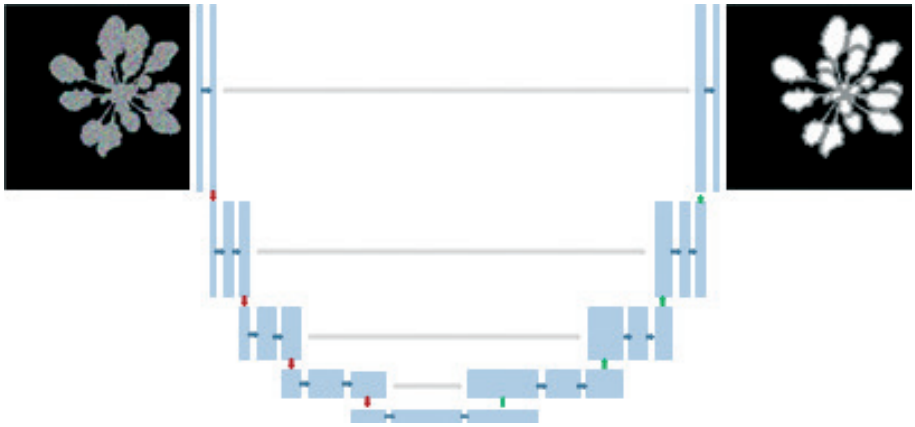


Figure 3.15 – U-Net architecture. Each blue box corresponds to a multi-channel feature map. The input image has 128x128 pixels, the output of the model is a three-channel binary image: mask without contours, leaf contours, and background.

Data augmentation

In order to produce some data augmentation, we consider binary images such as the ones in Figure 3.19 column (b) and map a noisy texture learned from the real fluorescence images shown in Figure 3.19 column (a). A copy of the original binary image for each plant is also kept to produce the associated ground-truth.

As the first trial of transfer from RGB images to fluorescence images, we propose to test a straightforward model for the noisy texture, estimated as an additional Gaussian white noise process that is independent and identically distributed for a given leaf. This choice is driven both by an Occam razor simplicity spirit. Indeed with such a model, the simulated leaves have no spatial structures such as vascular veins. Leaves are therefore expected to be differentiable in real images only from there first-order statistics. Also, as an additional motivation to test this simple fluorescence chlorophyll simulator, the noise in real fluorescence images is expected to be mostly thermal noise on the camera which will control the standard deviation of the noise and the leaves themselves if considered has homogeneous tissue may have a variety of reflectance depending on their physiological state.

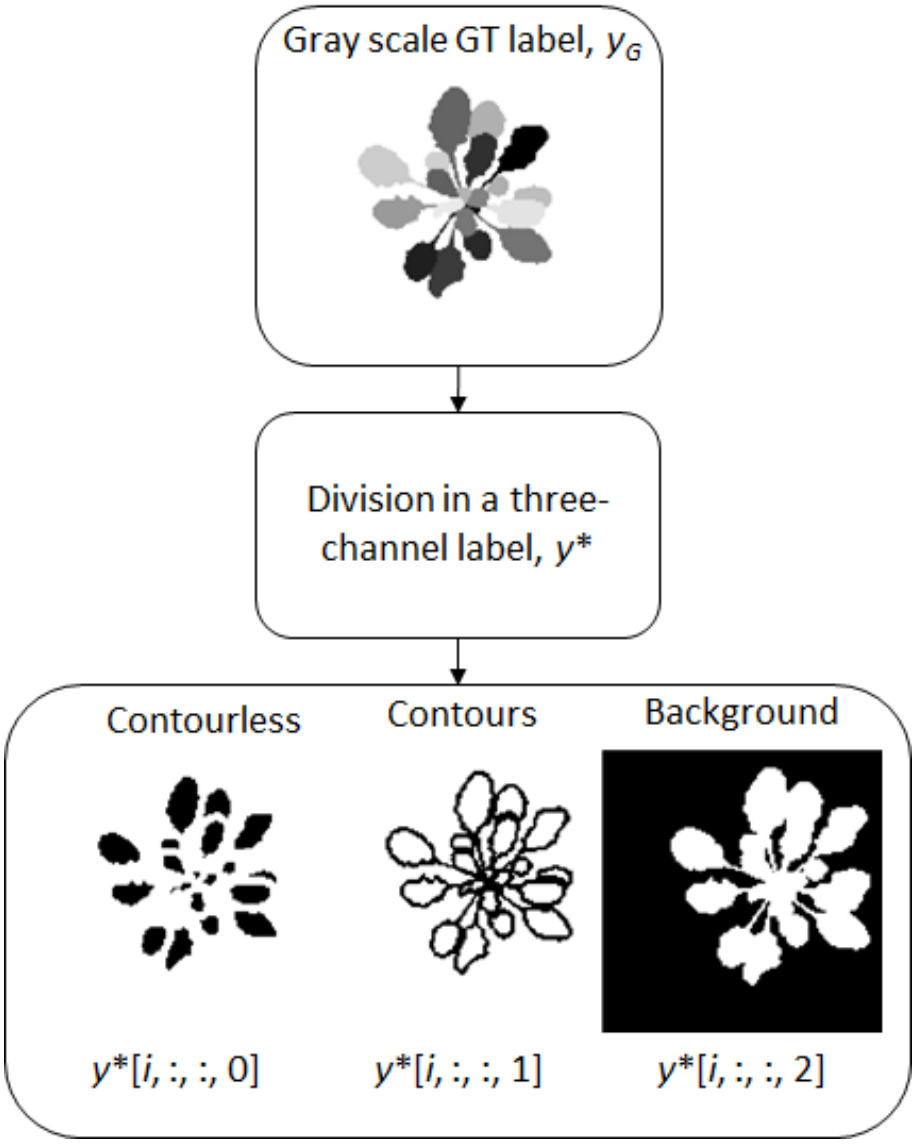


Figure 3.16 – Production of the three-channel binary labels from ground-truth (GT) label: the first channel contains mask without leaf contours, the second - leaf contours, and the third one - background.

We analyzed the distribution of the gray levels among a small set of images of real plants to estimate the parameters of these Gaussian processes. In order to ensure that this small set of chlorophyll fluorescence images is representative from the rest of the images, we considered one image of the plant at each developmental stage represented in the test dataset. The estimated average value and standard deviation of the gray levels inside the plant for both considered chlorophyll fluorescence parameters are given in Table 3.4.

The order of magnitude of the average value and standard deviation of the chlorophyll fluorescence parameters F_0 and F_m remains in the same order of magnitude.

Synthetic chlorophyll fluorescent images are then simply produced by adding Gaussian noise with distinct seven parameter sets $(\mu_{F_0}, \sigma^2_{F_0})$, $(\mu_{F_m}, \sigma^2_{F_m})$ to every label from *CVPPP* and *CSIRO* datasets:

$$x_F = 1 - \frac{y_g + n(\mu_{F_0}, \sigma^2_{F_0})}{y_g + n(\mu_{F_m}, \sigma^2_{F_m})}, \tag{3.12}$$

where x_F is a synthetic fluorescent image, y_g is a gray scale label and $n(\mu_{F_0}, \sigma^2_{F_0})$ is a Gaussian noise function. Values for μ_{F_0}, σ_{F_0} and μ_{F_m}, σ_{F_m} are randomly chosen among the values of Table 3.4. The pipeline of data augmentation is shown in Figure 3.17. As a result, we obtained new datasets, *CVPPP Fluo* and *CSIRO Fluo*, containing 5670 and 70000 synthetic fluorescent images (width x height: 128 x 128 pixels), respectively. Our objective is now to compare the added value of these datasets for leaf segmentation with the U-Net model presented in the previous section.

Time	μ_{F_0}	σ_{F_0}	μ_{F_m}	σ_{F_m}
Day 1	167.83	34.88	180.77	24.68
Day 5	165.81	33.1	180.00	22.36
Day 6	164.48	30.87	177.9	20.8
Day 7	158.16	31.45	174.73	21.1
Day 8	165.24	32.31	181.14	21.36
Day 9	168.3	28.03	184.36	17.86
Day 12	173.06	28.01	189.96	17.15

Table 3.4 – Mean, μ, and standard deviation, σ, for measurements of chlorophyll fluorescence: F_0 - minimal fluorescence, F_m - maximal fluorescence. Each line corresponds to an *Arabidopsis* after the indicated day following deployment of cotyledons.

Watershed post-processing

To segment leaves with the use of estimated 3D labels, we applied the marker-controlled watershed segmentation [326, 327]. The watershed concept is one of the standard tools in the field of topography. It is the line that determines where a drop of water will fall into a particular region. In mathematical morphology, gray-scale images are considered as topographic surface. If we flood this surface from its minima and prevent merging of the waters coming from different sources, we effectively partition the image into different segments, thereby revealing ridges. Flooded basins correspond to homogeneous regions

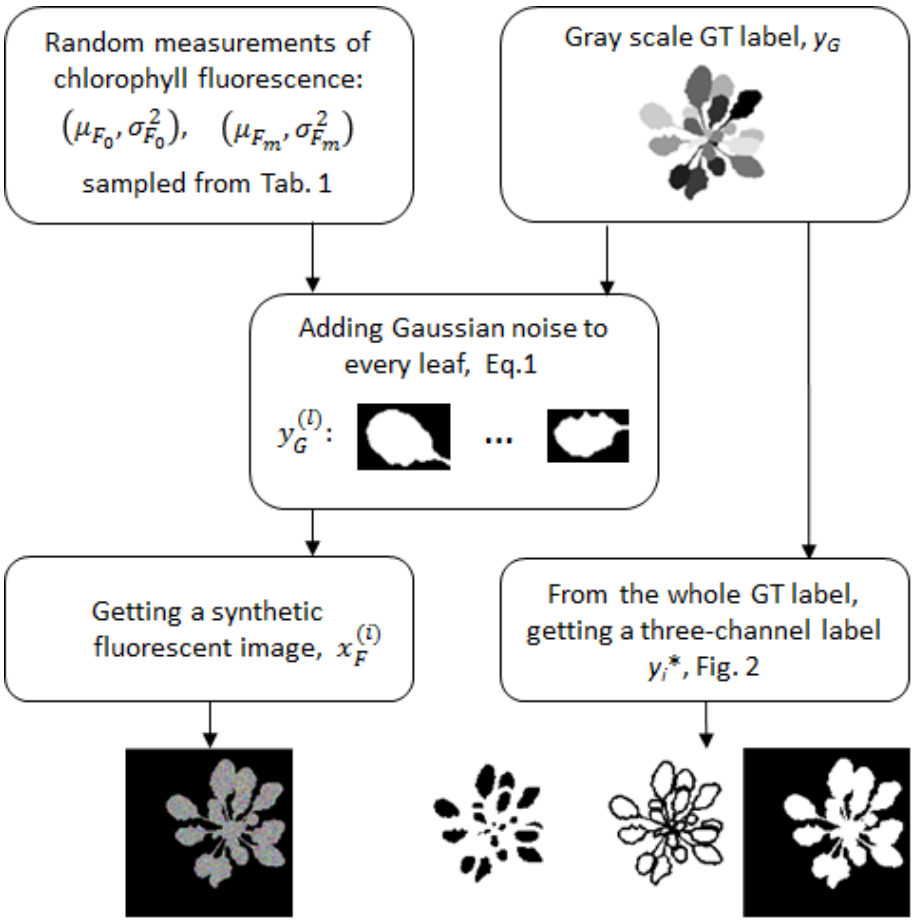


Figure 3.17 – Getting synthetic fluorescent training data. For each gray-scale label from the original dataset we produce seven fluorescent images and seven 3D labels.

in the image. If they are marked such that each marker is placed inside a basin under a one-to-one relationship, the watershed transform can segment regions with closed contours. To generate the markers, we used a contourless mask from output three-channel label, and then, to segment leaves, we flooded marked basins within the mask's bounds. Figure 3.18 illustrates the resulting leaf segmentation for different training strategies.

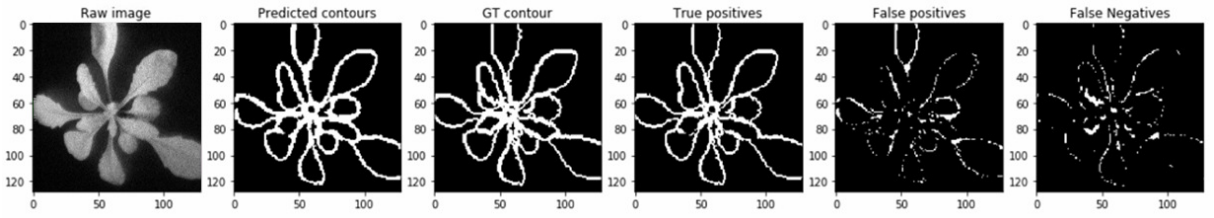


Figure 3.18 – Leaf segmentation results for different training strategies.

Training process

A standard data augmentation strategy was used on the input images from the different datasets shown in Figure 3.19 in order to reduce overfitting and improve generalization. For data augmentation, we used the Albumentations library [231]. Horizontal flip, vertical flip, random brightness, random contrast, random rotate at 90 degree; the random half-sized crop was applied to 0.7 shuffled training dataset.

It was shown that for the high level of imbalance, loss functions based on overlap measures appeared to be more robust [328]. Through all of our experiments, we minimized weighted combination of multi-class cross-entropy and dice losses:

$$\begin{aligned} L(y, y^*) = w_0 C(y, y^*) + w_0(1 - D(y[..., 0], y^*[..., 0])) \\ + w_1(1 - D(y[..., 1], y^*[..., 1])). \end{aligned} \tag{3.13}$$

$C(y, y^*)$ is the categorical cross entropy defined as:

$$C(y, y^*) = -\sum_{ij} y_{ij} \log y^*_{ij} \tag{3.14}$$

and $D(y, y^*)$ is the Dice coefficient:

$$D(y, y^*) = \frac{2 \sum_{ij} y_{ij} y^*_{ij} + \epsilon}{\sum_{ij} y_{ij} + \sum_{ij} y^*_{ij} + \epsilon}, \tag{3.15}$$

where y is a model prediction with values y_{ij}, y^* is a ground-truth label with values y^*_{ij} and ϵ is used here to ensure the coefficient stability by avoiding the numerical issue of dividing by 0. The weights ratios used to correct the class imbalance were, respectively, at 0.4, 0.1, and 0.5 for cross-entropy, contourless masks, and contours. Adam optimizer was used with default parameters $lr = 0.001, beta_1 = 0.9, beta_2 = 0.999$. Our training procedure consisted of splitting the data into 80% and 20% training and cross-validation, respectively. We shuffled the dataset examples at the beginning of each epoch and used a batch size of 16 examples. We have also implemented batch normalization before each activation.

Different training strategies to predict the fluorescence images with different datasets were tested for comparison. A baseline consists of training directly on the CVPPP RGB images. The learning from the simulated fluorescence dataset either generated from CVPPP labels and/or CSIRO labels is also tested. The previous strategies are tested

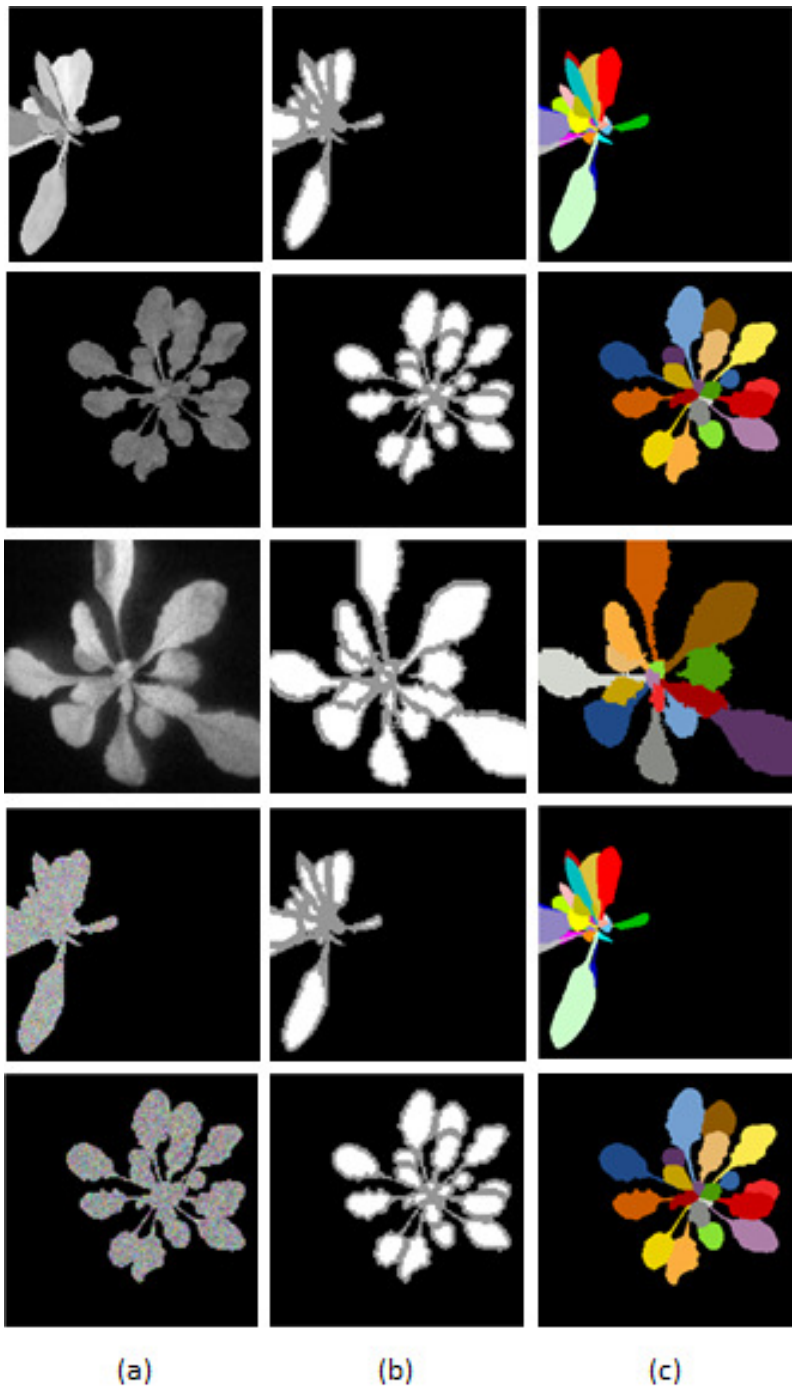


Figure 3.19 – datasets used for model training and its evaluation. (a) Plant image examples. (b) Three class labels for pixel-wise classification. (c) Ground-truth labels with leaf segmentation.

even a small number of real fluorescence images are added in training. The eight different tested training strategies are summarized in Table 3.5.

Results

To assess the quality of segmentation, we used the soft Dice coefficient Eq. (3.15). Table 3.5 displays the model performance on the *Real Fluo* dataset for the eight model training experiments of Table 3.4.

Training dataset	Loss_{train}	DiceCoeff_{train}	Loss_{test}	DiceCoeff_{test}
CVPPP	0.03	0.98	0.19	0.945
CVPPP Fluo	0.01	0.99	0.29	0.94
CSIRO Fluo	0.018	0.99	0.27	0.92
CVPPP Fluo + CSIRO Fluo	0.011	0.99	0.297	0.924
CVPPP + Real Fluo 10ex	0.036	0.975	0.049	0.973
CVPPP Fluo + Real Fluo 10ex	0.015	0.99	0.05	0.97
CSIRO Fluo + Real Fluo 10ex	0.026	0.98	0.062	0.96
CVPPP Fluo + CSIRO Fluo + Real Fluo 10ex	0.037	0.97	0.054	0.962

Table 3.5 – Performance in terms of the loss function and Dice of our leaf segmentation system on testing datasets with the various data augmentation technique tested.

As visible in Table 3.5, predicting Fluorescence from RGB images already provides a good segmentation baseline, which overpasses the simulation from fluorescence images.

The best model Dice score is 97% obtained for extended *CVPPP Fluo* dataset with 10 examples from *Real Fluo* dataset. The use of a small quantity of real fluorescent images among images with modeled fluorescence resulted in a score gain of 3% compared with *CVPPP Fluo* dataset. We observed the same positive effect of the injection of 10 real fluorescent images on the Dice score for the other datasets. The comparison of scores from *CVPPP* and *CVPPP Fluo* showed that the imitation of fluorescence by modeling did not have any impact on the quality of leaf contour segmentation for real plant fluorescent images.

Conclusion and discussion

In this work, we study the performance of the transfer of leaf segmentation learned from RGB imaging modality to fluorescence modality. We have shown that simple modeling of the noise in fluorescence imaging as a Gaussian noise is valuable enough to simulate data that can improve the segmentation of leaves on real data. This is shown efficient

both with RGB images from real plants or simulated plants. The gain found is, of course, higher when some real images are also introduced in the training process.

3.4 Computationally light deep architectures

Deep learning is currently tested world-widely in almost all application domains of computer vision as an alternative to purely handcrafted image analysis [329]. When inspecting the convolutional coefficients in first layers of deep neural networks, these are very similar to Gabor wavelets. While promoting a universal framework, deep neural networks seem to systematically converge toward tools that humans have been studying for decades. This empirical fact is used by computer scientists in the so-called transfer learning where the first layers of an already trained network are re-used [330]. This has also triggered interest by mathematicians to revisit the use of wavelets to produce universal machine learning architectures. This interdisciplinary cross-talk resulted in the proposal of the so-called scatter transform [331], which is roughly a cascade of wavelet decomposition followed by non-linear and pooling operators. If this deep architecture bares some similarity with the standard deep learning it does not include the time consuming feed-forward propagation algorithm. However, it proved its comparable efficiency to deep learning while offering a very rational way of choosing the parameters of the network compared to the rather empirical current art of tuning neural networks.

Despite its intrinsic interest to address multiple scales problems compared to deep learning, scatter transform since its introduction in 2013 has been applied only on a relatively small variety of pattern recognition computer vision problems notably including iris recognition, [332] rainfall classification in radar images, [333] cell-scale characterization, [334, 335] or face recognition, [336]. Also, in these applications scatter transform has shown its efficiency but it was not systematically compared with other techniques in a comprehensible way. We propose to extend the scope of investigation of the applicability of scatter transform algorithm to plant science with the problem of weed detection in a background of culture crops of high density used for the contribution on speeding up image annotation with desktop eyetracking. From a methodological point of view, this classification problem here will also serve as a use case to assess the potential of the scatter transform when compared with other single scale and multiple scales techniques.

A large variety of platforms, sensors and data process already exist to monitor weeds at various temporal and spatial scales. From remote sensing supported by satellites to

cameras located on unmanned aerial vehicles (UAVs) or on ground-based platforms, many systems have been described and compared for the weed monitoring in arable culture crops [337, 338, 339]. Related to the observation scale of our use case, by focusing on the imaging scales of UAVs and ground-based platforms, some studies exploiting RGB data have addressed crop weed classification with a large variety of machine learning approaches. The problem of segmentation of crop fields from typical weeds, performing vegetation detection, plant-tailored feature extraction, and classification to estimate the distribution of crops and weeds has recently been solved with convolutional neural networks in the field [340, 341] and in real-time [342]. Earlier, Aitkenhead, M. et al. [343] evaluated weed detection in fields of crop seedlings using simple morphological shape characteristic extraction and self-organizing neural network. Bayesian classifier was used in [344] for plant and weed discrimination. Shape, texture features [345, 346, 340, 347] or wavelet transform [348, 349] coupled with various classifiers including support vector machine (SVM), relevance vector machine (RVM), fuzzy classifier or random forests were also shown to provide successful pipelines to discriminate between plant and weeds.

The above list of reference is of course not exhaustive and new pipelines will continue to appear because of the large variety of crops shape and imaging platform. In this context, scatter transform constitutes a candidate of possible interest worth to be assessed on a plant–weed classification problem. Also, by comparison with the existing work on weed detection, the computer vision community has focused on the relatively low density of crops and weed where the soil constitutes a background to be classified in addition to crop and weed. In this section, we consider the case of culture crops of high density, i.e. where the soil is not visible from the top view. In this case, the culture is the background and the object to be detected are weeds of wild type. The contrast in color between the background and the weed, in this case, is obviously here very low by comparison with lower density culture

Since the data set and the computer vision problem has been presented earlier in the human-assisted annotation section, we directly present the expected scales included in the images and the algorithms tested for comparison with the multiscale scatter transform algorithm.

Scales

With a spatial resolution of 5120 by 3840 pixels included in the images of our dataset, and as illustrated in Fig. 3.20, multiple anatomical structures of the dense weed/plant

culture are accessible in our images. From tiny to coarse sizes, i.e. scales, this includes texture in the limb, the veins, and the leaf. There are possibly discriminant features between the two classes (weed/plant) to be found in these three scales either taken individually or combined with each other. To offer the possibility of a multiple scale analysis, together with a reasonably small computation time, classification is done at the scale of patches chosen as double size of the typical size of leaves, $2 \times max\{S_w, S_p\}$, with rectangles of 250 by 325 pixels where $S_w = 163$ pixels and $S_w = 157$ in average. With this constraint, we also keep for the patch the same ratio between height and width as in the original image for a periodic patch grid.

dataset

With the simulator of Fig. 3.5, we produced a total amount of 3292 patches containing weed and 3292 patches only with plants. The binary classification (weed/plant) is realized on these patches. This balanced dataset serves both for the training and the testing stages to assess the performance of different machine learning tools. The datasets together with the simulator are proposed as supplementary material under the form of a free executable and a set of images [1].

Classifiers

In this section, we describe how we apply the scatter transform [331] on the weed detection problem introduced in the previous section. For comparison, we then propose a set of alternative techniques.This study uses independent k-fold cross-validation to measure the performance of the scatter transform coupled to the classifier depicted in Fig. 3.21 and compare other feature extractors coupled to the same classifier. The performances of these classifiers are measured by the metric of the accuracy of correct classification by

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.16}$$

where TP indicates that the prediction is positive and the actual value is positive. FP indicates that the prediction value is positive but the actual value is negative. TN indicates that the prediction value is negative and the actual value is negative. FN indicates that the prediction value is negative but the actual value is positive.

1. https://uabox.univ-angers.fr/index.php/s/iuj0knyzOUgsUV9

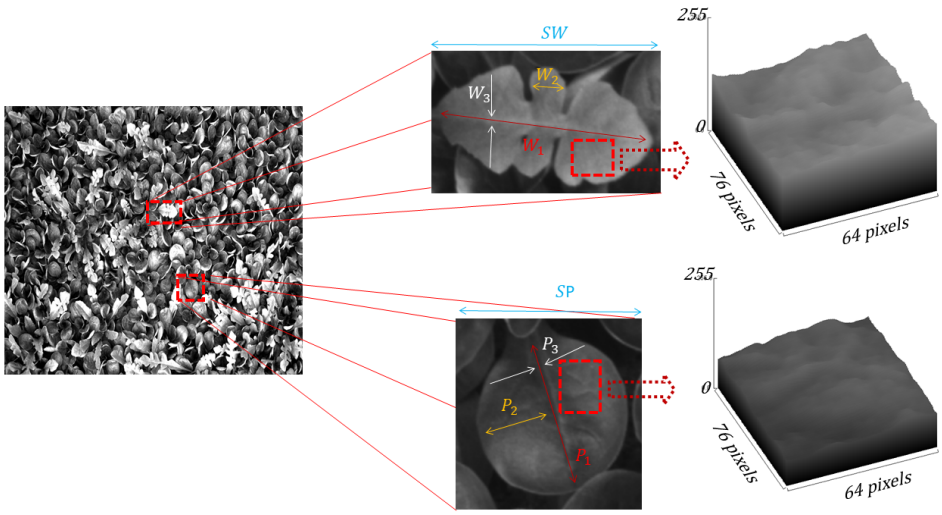


Figure 3.20 – Anatomical scales where (W_i,P_i) presents the scales of weeds and plants respectively; (W_1, P_1) points toward the texture of the limb, (W_2, P_2) indicates the typical size of leaflet and (W_3, P_3) stands for the width of the veins. Sw and Sp show the size of a leaf of weed and plant respectively. The classification of weed and plant is done at the scale of a patch taken as $2 \times \max(Sp, Sw)$ in agreement with a Shannon-like criteria.

Scatter transform

A scattering transform defines a signal representation which is invariant to translations and potentially to other groups of transformations such as rotations or scaling. It is also stable to deformations and is thus well adapted to image and audio signal classification. A scattering transform is implemented with a convolutional network architecture, iterating over wavelet decompositions and complex modulus. Figure 3.21 shows a schematic view of a scatter transform network working as a feature extractor and coupled to a classifier after dimension reduction.

The scatter vectors Z_m at the output of the first three layers $m = 1, 2, 3$ for an input image f are defined by

$$\begin{aligned} Z_1 f &= \{|f| \star \phi\} \\ Z_2 f &= \{\dots, |f \star \psi_{j,\theta}| \star \phi, \dots\} \\ Z_3 f &= \{\dots, ||f \star \psi_{j,\theta}| \, \psi_{k,\varphi}| \star \phi, \dots\} \ , \end{aligned} \tag{3.17}$$

where the symbol $\star$ denotes the spatial convolution, $|.|$ stands for the L_1 norm, ϕ is an averaging operator, $\psi_{j,\theta}$ is a wavelet dilated by 2^j and rotated by θ. The range of scales

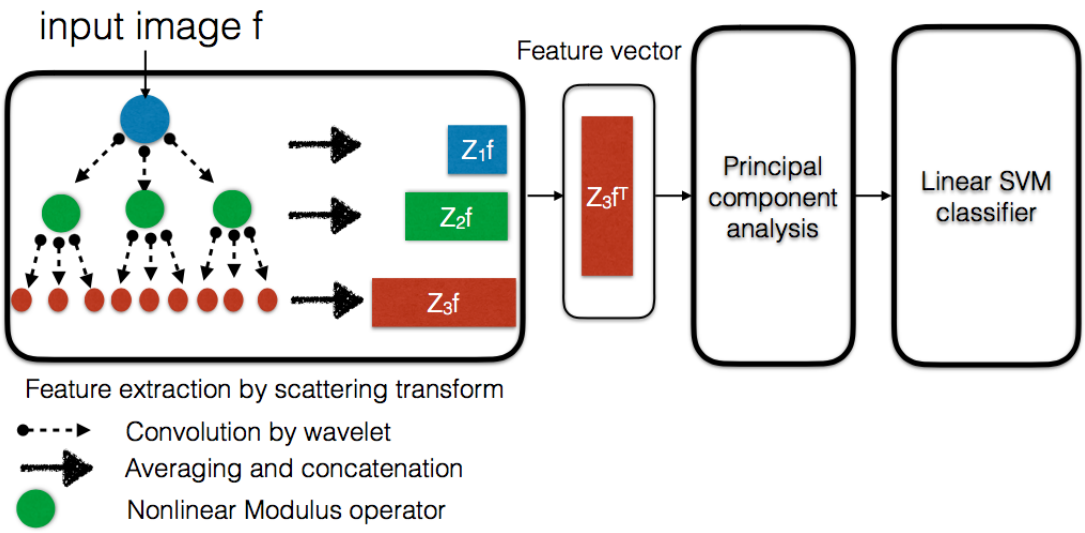


Figure 3.21 – Schematic layout of the weed/plant classifier based on the scattering transform with three layers. The feature vector transmitted to the principal component analysis (PCA) step consists in the scatter vector $Z_m f$ of the last layer of Eq. (3.17) after transposition.

$j = \{0, 1, \ldots, J\}$ and the number of orientations $\theta = \{0, \pi/L, \ldots, \pi(L-1)/L\}$ are fixed by integers J and L. The number of layers is between $m = 1$ to $m = M$. In our case, we considered as mother wavelet the Gabor filter with implementation provided under MatLab in (https://www.di.ens.fr/data/scattering/) for scatter transform.

Scatter transform differs from a pure wavelet decomposition because of the nonlinear modulus operator. With this nonlinearity, decomposition of the image is not done on a pure orthogonal basis (whether wavelet basis is orthogonal or not) and this opens the way of a possible benefit in the concatenation of several layers with a combination of wavelet decompositions at different scales. Interestingly, these specific properties of the scatter transform match the intrinsic multiscale textural nature of our weed detection problem which therefore constitutes an appropriate use case to assess the potential of the scatter transform in practice. A visualization of output images for various filter scale j at $m = 2$ for a given orientation is shown in Fig. 3.22. It clearly appears in Fig. 3.22 that the various scales (texture of the limb and veins at j=3, border shape at j=4 and global leaf shape at j=8- not shown) presented in section 2.2 can be captured with the different scaling factor applied on the wavelet. In our study, we empirically picked $L = 8$ orientations and investigated up to $J = 8$ scales since there are no other anatomical items larger than the leaf itself. The number of layers tested was up to $M = 4$ as proposed in

[331] since the energy after some layers although none zero is logically vanishing.

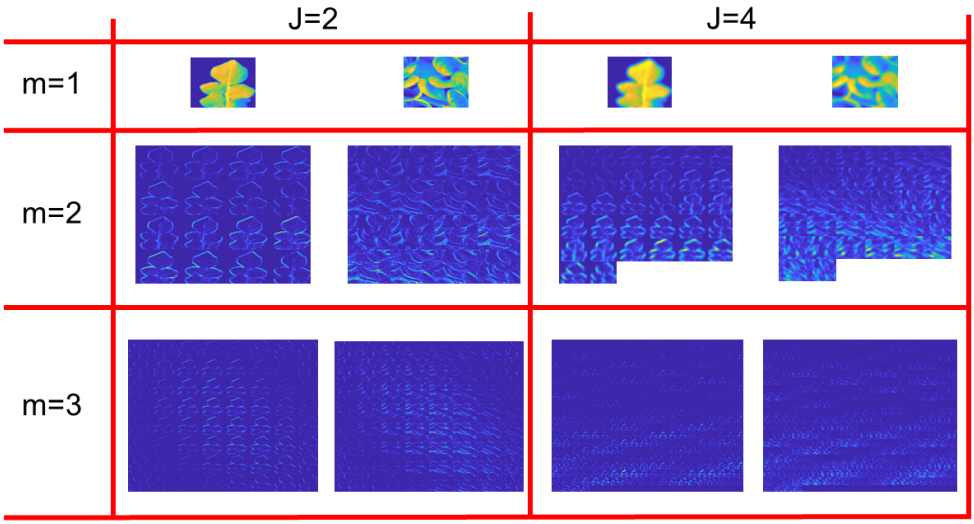


Figure 3.22 – Output images for each class (weed on left and plant on right) and for each layer m of the scatter transform.

In the application of scatter transform to classification found in the literature so far, the optimization of the architecture was done a posteriori after supervised learning. This is rather time-consuming. We investigated the possibility to select a priori the best architecture by analyzing the distribution of relative energy E_m at the output of each layer as given by

$$E_m = ||Z_m f|\,|^2 / \,||f|\,|^2 \,. \tag{3.18}$$

We computed these energies for the whole dataset as given in Table 3.6. As noticed in [331], the relative energy is progressively vanishing when the number of layers increases. This observation advocates for the use of a limited number of layers. However, these energies are computed on the whole population of patches including both plants and weeds and therefore it tells nothing about where to find the discriminant energy between each class throughout the feature space produced by the scatter transform. Tables 3.7 and 3.8 show the average relative energy for the weeds' patches data-set, $\overline{E}_{w_m}$, and plants' patches data-set,$\overline{E}_{p_m}$, for different layers m and various maximum scale J.

In order to show this discriminant energy between each class, various criterion could be proposed. We tested the percentage of energy similarity, Q_m, between the two classes

defined by

$$Q_m = \frac{argmin(\overline{E}_{w_m}, \overline{E}_{p_m})}{argmax(\overline{E}_{w_m}, \overline{E}_{p_m})} \times 100. \tag{3.19}$$

According to this criterion, the best architecture of the scatter transform can be chosen at the point of η where the minimum Q_m between each classes is found as a function of J by $\eta = argmin_J(Q_m(J))$. The energy similarity $Q_m(J)$ are represented in Fig. 3.23 and this clearly demonstrates that the contrast between classes is more pronounced on coefficient with small relative energy. This observation, not stressed in the original work of [331], indicates that it should be possible to draw benefit from the contribution of these small discriminative coefficients and thus this demonstrates the interest of the combinatory step of the scatter transform.

	m=0	m=1	m=2	m=3	m=4
J=1	96.18	2.35	-	-	-
J=2	91.81	4.61	0.28	-	-
J=3	85.81	8.46	0.89	0.03	-
J=4	85.81	13.15	1.97	0.17	0.006
J=5	81.46	15.36	3	0.36	0.024
J=6	79.04	16.81	3.44	0.53	0.048
J=7	80.74	17.05	3.49	0.63	0.071

Table 3.6 – Average percentage of energy of scattering coefficients E_m on frequency-decreasing paths of length m (scatter layers), with $L = 8$ orientations and various filter scale range, J, for the whole database of plants and weeds patches.

	m=0	m=1	m=2	m=3	m=4
J=1	99.90	0.0985	-	-	-
J=2	99.71	0.2798	0.0098	-	-
J=3	99.07	0.8832	0.0443	0.0016	-
J=4	97.55	2.2669	0.1663	0.0080	0.0003
J=5	95.10	4.3892	0.4667	0.0343	0.0020
J=6	92.07	6.8696	0.9522	0.0983	0.0076
J=7	89.26	9.0102	1.5049	0.1979	0.0196

Table 3.7 – Average percentage of energy of scattering coefficients E_m on frequency-decreasing paths of length m (scatter layers), depending upon the maximum scale J and $L = 8$ filter orientations for the weed class patches.

Also, from the observation of Fig. 3.23, our approach indicates that a priori the best discriminant energy between each class is to be expected with a scatter architecture

	m=0	m=1	m=2	m=3	m=4
J=1	99.92	0.0711	-	-	-
J=2	99.76	0.2339	0.0040	-	-
J=3	99.17	0.7984	0.0281	0.0003	-
J=4	97.75	2.0899	0.1380	0.0041	0.00003
J=5	95.41	4.1411	0.4215	0.0254	0.0006
J=6	92.34	6.6553	0.9078	0.0892	0.005
J=7	89.37	8.9341	1.4817	0.1944	0.0171

Table 3.8 – Average percentage of energy of scattering coefficients on frequency-decreasing paths of length m (scatter layers), depending upon the maximum scale J and $L = 8$ filter orientations for the plant class patches.

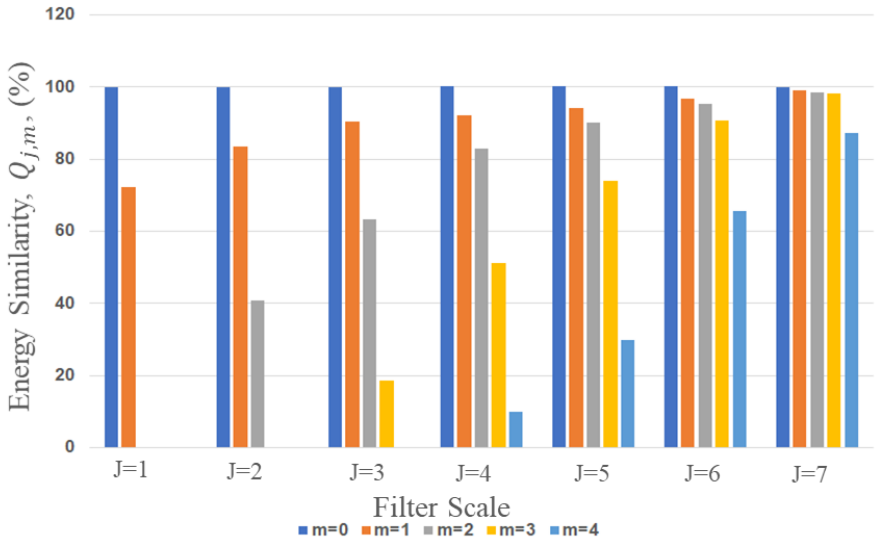


Figure 3.23 – Energy similarity, $Q_m(J)$, between energy of weeds and plants datasets based on Tables 3.8 and 3.7.

corresponding to $M = 4$ and $J = 4$ which provides the minimum energy similarity, η, between the energy of images of the weeds' class and the plants' class.

Other methods

To assess the possible interest of the scatter transform in our weed detection problem, we consider several alternative feature extractor algorithms. First, since the scatter transform by construction works on a feature space which includes multiple scales, it is expected to perform better than any state-of-the-art mono-scale method, i.e. working on a feature space tuned on a single size, when applied on a multiple scales problem (such as the one we have here with veins, limb, leaf). Second, since the scatter transform works on a combination of wavelet decomposition between scales it should perform slightly better than a pure wavelet decomposition chosen on the same wavelet basis but without the use of the non-linear operator nor the scales combination. Last but not least, because scatter transform shares some similarities with convolutional neural networks it should also be compared with the performance obtained with a deep learning algorithm. Based on this rationale, we propose LBP, GLCM, and Gabor filter as feature extractor for comparison with the feature extractor of the scatter transform where the same PCA followed by a linear SVM is used for the classification. In addition to these shallow learning methods we add several deep learning methods of various computational costs.

Deep learning

Representation learning, or deep learning, aims at jointly learning feature representations with the required prediction models. We chose the predominant approach in computer vision, namely deep convolutional neural networks (ConvNets) [350]. The baseline approach resorts to standard supervised training of the prediction model (the neural network) on the target training data. No additional data sources are used. In particular, given a training set comprised of K pairs of images f_i and labels $\hat{y}_i$, we train the parameters θ of the network r using stochastic gradient descent to minimize empirical risk:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{K} \mathcal{L}(\hat{y}_i, r(f_i, \theta)) \tag{3.20}$$

$\mathcal{L}$ denotes the loss function, which is cross-entropy in our case. The minimization is carried out using the ADAM optimizer [66] with a learning rate of 0.001. The architecture of network $r(\cdot, \cdot)$, shown in Fig. 3.24, has been optimized on a hold-out set and is given as follows: five convolutional layers with filters of size 3×3 and respective numbers of filters 64, 64, 128, 128, 256 each followed by ReLU activations and 2×2 max pooling; a fully connected layer with 1024 units, ReLU activation and dropout (0.5) and a fully connected

output layer for 2 classes (weeds, plants) and softmax activation. Given the current huge interest on deep learning many other architectures could be tested and possibly provide better results. As a disclaimer, we stress that the architecture proposed in Fig. 3.24 is of course not expected to provide the best performance achievable with any neural network architecture. Here the tested CNN serves as a simple reference with a level of complexity of the architecture adapted to the size of the input image and training datasets.

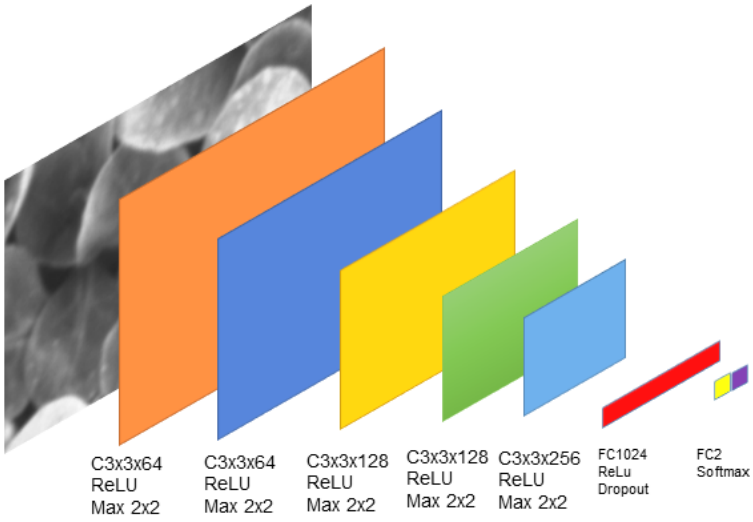


Figure 3.24 – Architecture of the deep network optimized for the task on classification.

Light deep architecture - TensorFlow Lite

As machine learning tasks are computationally expensive, model optimization is used to reduce the computation cost and improve performance. Lightweight solutions allow us to convert the model trained on a higher-powered machine, to a light version to use on mobile and embedded devices. Google developed TensorFlow [351] as a machine learning system that operates on a large scale and in heterogeneous environments, [2] including low-power edge technologies. However, because of latency due to the problem of sending data back and forth between devices and data centers, Google designed TensorFlow Lite (TFL), a framework for machine learning inference on embedded devices, which computations could be executed on the device without the need for network round-trip delays. TFL is considered as an optimized solution to the deep neural network's bottleneck, which is

2. https://www.tensorflow.org/

the calculation speed of the computations, as the desired latency for mobile applications is low. Furthermore, in TFL, the minimum hardware requirements in terms of Random Access Memory (RAM) size and CPU speed are low [352].

There is a possibility of using available pre-trained models or, like what is done in this study, using the trained model on the dataset associate with the task. We take benefit of TFL potential to design a network with lower computationally-demanding. For this reason, the trained models on different numbers of sample data explained beforehand are used. Then, in consideration of optimizing, these trained models are converted to the ".tflite" models automatically by TensorFlow and tested on a new dataset. The recognition accuracy on this approach is approximately similar to the baseline convolution neural network, as it is shown in Fig. 3.26.

Light deep architecture - MobileNet

Finally, we tested another practical model called MobileNet [353], which is an efficient convolutional network for low-power embedded systems. The first version of MobileNet uses depthwise separable convolutions to build light weight deep neural networks. Depthwise separable convolution is a form of factorized convolutions that factor a standard convolution into a depthwise convolution that applies a single filter to each input channel, and a 1×1 convolution called a pointwise that applies a 1×1 convolution to combine the outputs of the depthwise convolution by keeping the number of channels the same or doubled them. This factorization has the effect of remarkably reducing computation and model size. The full architecture of MobileNet V1 consists of a regular 3×3 convolution as the very first layer, followed by 13 times the above building block.

The second version of MobileNet architecture [354] also uses depthwise separable convolutions. However, three convolutional layers include a 1×1 convolution to expand the number of data channels before it goes into the depthwise convolution, which is called the expansion layer—followed by a depthwise convolution that filters the inputs and—continued by a 1×1 pointwise convolution layer, which makes the number of channels smaller by reducing the amount of data that flows through the network. This is why this layer is known as the projection layer, which is the opposite of the expansion layer. MobileNet version 2 also consists of a residual connection which helps with the flow of gradients through the network. Figure 3.25 shows the structure of the standard convolutional network and both versions of MobileNet architectures.

We use the MobileNet version 2 and re-train the classifier to detect the weeds on

the dense plants on our dataset to build a smaller and faster network by trading off a reasonable amount of accuracy to reduce size and latency, as shown in Fig. 3.26.

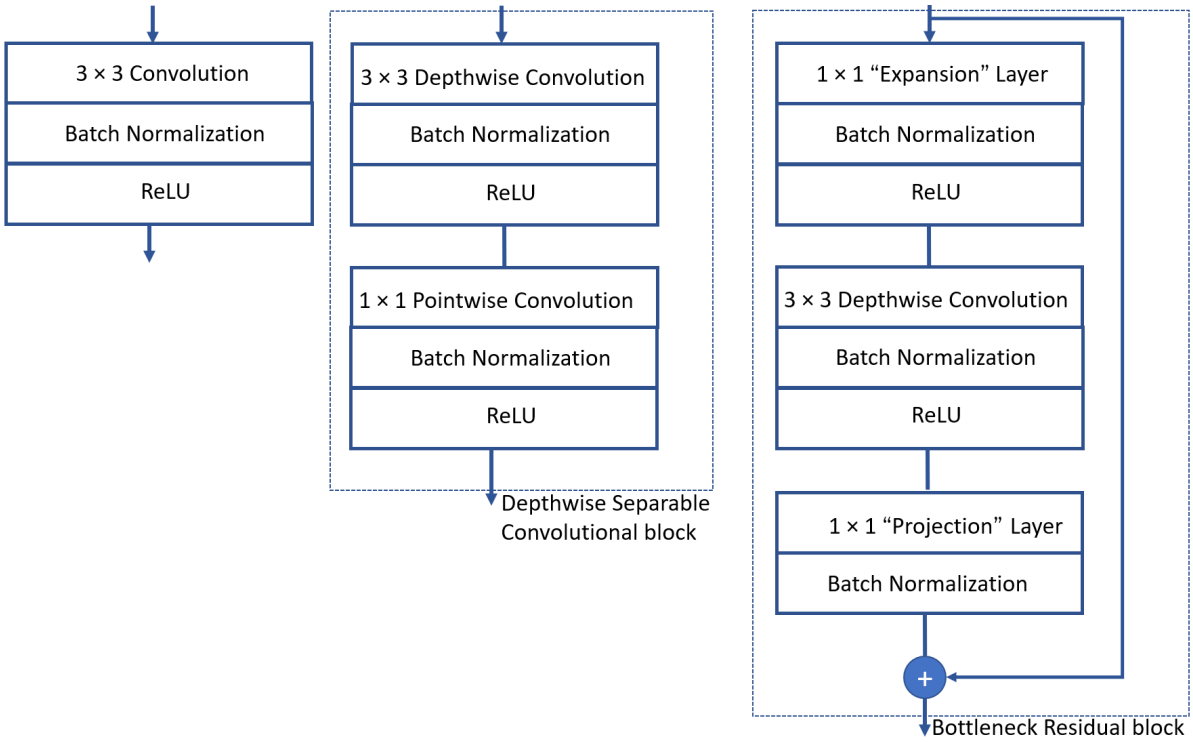


Figure 3.25 – Left: Standard convolutional network architecture with batch normalization and non-linearity. Middle: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batch normalization and non-linearity. Right: Expansion block consists of expansion layer with batch normalization and non-linearity followed by the depthwise block and the pointwie block including the projection layer and normalization and the residual connection.

Result

In this section, we provide experimental results using the experimental protocol for the assessment of scatter transform (section 3.4) as well as the different alternative feature extraction techniques chosen for comparison in section 3.4.

The scatter transform produces a data vector containing the $Z_m f$ of Eq. (3.17) whose dimension is reduced by a standard PCA and then applied to a linear kernel SVM. In order to compare the performance of different structures of scatter transform on the database, we used a different combination of filter scales, j, and the number of layers, m, to realize

which structure is the best fit for our data. Table 3.9 shows the classification accuracy of these structures where 10-fold cross-validation approach is used for classification. The best weed/plant classification results with scatter transform are obtained for $J = 4$ and $m = 4$. This a posteriori exactly corresponds to the prediction done a priori from the energy-based approach presented in the method section.

	J=1	J=2	J=3	J=4	J=5	J=6	J=7	J=8
m=1	70.37%	77.89%	82.74%	86.17%	88.96%	91.94%	94.14%	95.05%
m=2	—-	91.95%	95.26%	95.54%	95.86%	95.82%	95.73%	95.55%
m=3	—-	—-	95.41%	95.44%	95.21%	95.07%	95.03%	96.00%
m=4	—-	—-	—-	96.31%	96.02%	96.05%	96.16%	96.11%

Table 3.9 – Percentage of correct classification for 10 fold cross-validation classification on simulation data with scatter transform for various values of m and J.

We considered this optimal scatter transform structure with $J = 4$ and $m = 4$ and compared it with all alternative methods described in section 3.4. Table 3.10 shows the recognition rates of weed detection on the data where a k-fold cross-validation approach of SVM classification with the different number of folds is used. Scatter transform appears to outperform all compared handcrafted methods. This demonstrates the interest of the multiscale and combinatory feature space produced by scatter transform. It is important to notice that in order to have a fair comparison of these alternative methods, we adapted the feature spaces of all algorithms to the same size. The minimum size of the whole feature space is selected and feature space of other algorithms are reduced to that specific size. In our techniques, the minimum feature space belongs to the GLCM method which has a size of $N \times 19$ where N represents the number of samples. The PCA algorithm is adapted to our models to reduce the dimensions of the feature space generated by other techniques to the size of $N \times 19$.

As shown in Fig.3.26, when compared with deep learning approaches, like most handcrafted methods, scatter transform performs better for small datasets. The limit where deep learning and scatter transform are found to perform equally is found to be in a range of 3000 to 4500 based on deep learning approaches on the weed detection problem as given in Fig. 3.26. This demonstrates the interest of the scatter transform in case of rather small datasets. It is, however, to be noticed that an intrinsic limitation of scatter transform is that it works only with patches to perform a classification while some architectures of convolutional neural network would also be capable of performing segmentation directly in the whole image (see for instance U-Net) [323].

	5 Folds	6 Folds	7 Folds	8 Folds	9 Folds	10 Folds	Average std
Scatter Transform	94.9%	95.2%	95.3%	95.7%	95.8%	95.8%	± 1.1
LBP	85.5%	86.1%	86.3%	85.8%	86.9%	86.7%	± 0.4
GLCM	87.4%	91.6%	90.9%	92.1%	92.4%	92.3%	± 0.7
Gabor Filter	88.0%	88.2%	88.7%	88.6%	89.4%	89.3%	± 1.3
Deep Learning	89.4%	89.9%	91.1%	91.5%	91.9%	92.1%	± 1.4

Table 3.10 – Percentage of correct classification by using k-fold Cross-validation on 1200 simulated samples.

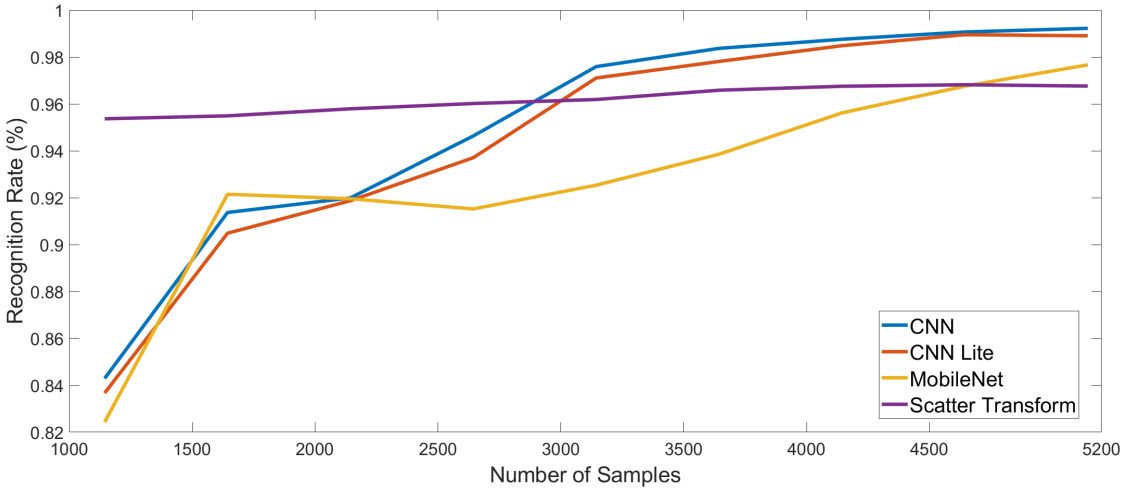


Figure 3.26 – Comparison of the recognition accuracy between scatter transform and baseline CNN plus CNNLite and MobileNet architecture when the number of samples increases.

Discussion

So far, we focused in this scientific research on detection of weeds in fields by the scatter transform algorithm with a comparison of other machine learning techniques which have been trained and tested on synthetic images produced by the simulator of Fig. 3.5. Our experimental results show that a good recognition rate of weeds detection (approximately 95%) can be achievable by the scatter transform algorithm. On the other hand, other alternative methods also work well for this problem with a minimum recognition rate around 85%. These experiments prove that texture based algorithms can be useful for weed detection in culture crops of high density.

One may wonder how these classification results compare toward the literature on weed detection in less dense culture cited in the introduction section [341, 342, 343, 344,

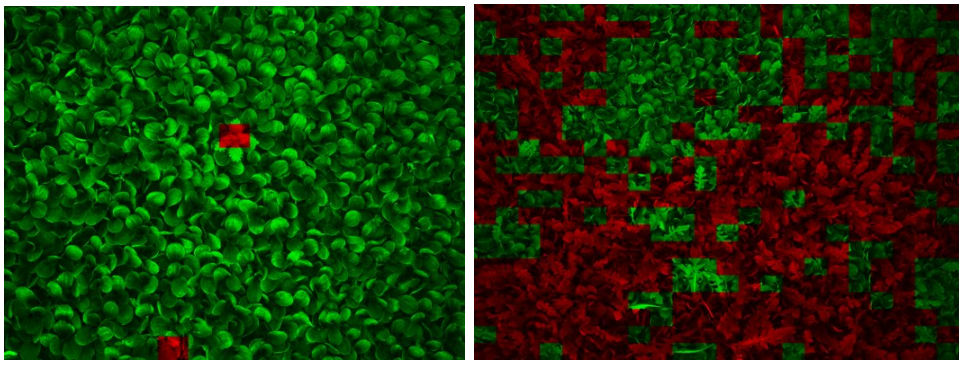
345, 346, 340, 347, 348, 349]. The performance in this literature varies from 75% to 99% of good detection of weed. It is, however, difficult to provide a fair comparison since in addition to the main difference with the absence of soil, the observation scales together with the acquisition conditions vary from one study to another.

One may wonder how these algorithms trained on synthetic data behave when they are applied to real images including plant background and weed not included in the synthetic datasets. We also tested our scatter transform classifier which was trained on synthetic data when applied on the real images of Fig. 3.3. On average for all 10 real images, the accuracy found is 85.64%. Although this constitutes already interesting results, this indicates a bias between simulated data and real data. One direction could be to improve the realism of the simulator. In the version proposed here weeds were not necessarily acquired in the same lighting conditions as the plant. A simple upgrade could be to adapt the average intensity on the weed and the plant to compensate for this artifact or, since in plant and weed can indeed be of various intensity, to generate data augmentation with various contrast. However, simulators never exactly reproduce reality. Another approach to improve the performance of the training based on simulated data would be to add a step of domain adaptation after the scatter transform [355]. So far, the best and worst results obtained with scatter transform are given in Fig. 3.27. A possible interpretation for the rather low performance in 3.27b is the following. The density of weed in Fig. 3.27b is very high compared to the other images in the training dataset. As a consequence, the local texture in the patch may be very different from the one obtained when weeds appear as outliers. This demonstrates that the proposed algorithm, trained on synthetic data, is appropriate in the low density of weeds at an observation scale like the one chosen for the patch where plant serves as a systematic background.

These performances could be improved in several ways. First, a large variety of weeds can be found in Nature and it would be important to include more of this variability in the training datasets. Also, weeds are fast growing plants capable of winning the competition for light. Therefore high percentages of weed is expected to come with higher weeds than in very low percentage of the surface of weeds. This fact illustrated in Fig. 3.27 is not included in the simulator where weeds of a fixed size are randomly picked. Such example of enrichment of the training dataset and simulator could be tested easily following the global methodology presented in this study to assess the scatter transform. Finally, we did not pay much effort on denoising the data. The proposed data have been acquired with a camera fixed on an unmanned vehicle. Compensation for variation of illumina-

tion in the dataset, or inside the images, themselves or compensation for the possible optical aberration of the camera used could also constitute directions of investigation to improve the weed/plant detection. All the methods presented in this study (including scatter transform) have the capability to be robust to global variation of light intensity however the variation of light direction during the day may impact the captured textures. Increasing the dataset to acquire images at all hour of a working day or adding a lighting cabinet on the robot used would make the results even more robust [342, 356, 357, 358].

The problem of weed detection in culture crops of high density is an open problem in agriculture which we believe deserves the organization of a challenge similar to the one organized on Arabidopsis in controlled conditions [359] for a biology community. Such challenges contribute to improving the state-of-the-art as recently illustrated with the use of simulated Arabidopsis data to boost and speed up the training [360] in machine learning. This challenge is now open on the codalab platform [3] together with the effort of proposing real data and the simulator [4] developed for this study. These additional materials, therefore, contributes to the opening of the problem of weed detection in culture crops of high density to a wider computer vision community.



(a) Image 2 (97.27%) (b) Image 9(69.45%)

Figure 3.27 – Visual comparison of the best and the worst recognition of weeds and plants by scatter transform.

Conclusions and perspectives

In this study, we proposed the first application of the scatter transform algorithm to plant sciences with the problem of weed detection in a background of culture crops of

3. https://competitions.codalab.org/competitions/20075
4. https://uabox.univ-angers.fr/index.php/s/iuj0knyzOUgsUV9

high density. This open plant science problem is important for field robotics where the mechanical extraction of weed is a current challenge to be addressed to avoid the use of phytochemical products.

We assessed the potential of the scatter transform algorithm in comparison with single scale and multiscale techniques such as local binary pattern, gray level co-occurrence matrix, Gabor filter, convolutional neural network and lightweight architectures such as tensorFlow Lite and MobileNet. Experimental results showed the superiority of the scatter transform algorithm with a weed detection accuracy of approximately 95% over the other single scale and multiscale techniques on this application. Scatter transformed also appeared as an interesting alternative to deep learning for training data set smaller than 10^4 instances. Though the comparison was not intended to be exhaustive among the huge literature on texture analysis, the variety of tested techniques contributes to confirm the effectiveness of using the scatter transform algorithm as a valuable multiscale technique for a problem of weed detection and opened an interesting approach for similar problems in plant sciences. Finally, an optimization method based on energy at the output of the scatter transform has been successfully proposed to select a priory the best scatter transform architecture for a classification problem.

Concerning the weed-plant detection, our optimal solution with scatter transform can serve as a first reference of performance and other machine learning techniques could now be tested in the framework of the data challenge that we launched for this scientific research [5]. As a possible perspective of the investigation, one could further optimize the scatter transform classifier proposed in this study. For instance, the size of the grid could be fine-tuned or some hyper-parameters could be added with nonlinear kernels in the SVM step. Also, weed/plant detection was focused here on a binary classification since no distinction between the different weeds were included. In another direction, one could also envision to extend this work to a multiple types of weeds classification problem if more data were included.

5. https://competitions.codalab.org/competitions/20075

Chapter 4

Conclusion and Perspectives

4.1 Synthetic view of contributions

In this thesis, we investigated the possibilities of performing high-throughput imaging for plant phenotyping at low-cost on a set of biological questions. Our contributions [361, 362, 363, 265, 72, 364, 365, 366, 367] can be organized into two parts. The first part focused on how to reduce the cost of plant phenotyping at the sensor level. In this part, we have shown innovative use of mini-computers originally developed for educational purposes, associated with RGB and/or LiDAR cameras, originally developed for video game purposes, to monitor plants from the top view as individuals [367], or at a canopy level [72]. For this part, we have created an imaging platform from scratch with a network of 60 cameras capable of monitoring 50000 seedlings in parallel. With more convenient access to imaging systems (possibly at low-cost as stressed in the first part), the current bottleneck in plant phenotyping now corresponds to the development of image processing algorithms. Given the large variety of biological questions and variability of plant shapes, plant imaging seems to produce an overwhelming need for the design of algorithms which overpasses any human capability to design specific algorithms for each question. The second part addressed this issue and focused on reducing the cost of the image processing algorithm's design. In the era of machine learning-driven computer vision, unequaled performances are accessible with advanced algorithms such as deep learning, which stands as universal algorithms for solving image processing problems. The bottleneck in the development of image processing solutions is no more the design of the algorithms itself but in the cost of the computation devices and the time required for the creation of ground truth associated with the images to be processed. In this context, we have investigated the value of the scattering transform [362, 367], a recently introduced deep architecture that bears some similarities with deep learning without the need for massive computational resources nor large annotated data sets. We

have also investigated the possibility of performing automated image annotation with unsupervised machine learning in sequences of images [361]. We have demonstrated, for the first time to the best of our knowledge, the possibility to speed up annotation with ergonomic tools based on the capture of the gaze of the eye of the annotator [265, 364]. We have quantified gain of time up to 70 with such tools illustrated in various use cases. Last, we have demonstrated the possibility to speed up annotation by the use of synthetic data automatically annotated [363, 368, 365].

This thesis proposed an analysis on how to reduce the cost of plant phenotyping with a set of practical use cases. In the meanwhile, the ensemble of work covered a large spectrum of methods as recalled in Fig. 4.1. This thesis was conducted in the ImHorPhen research team in LARIS "Laboratoire Angevin de Recherche en Ingénierie des Systèmes" Université d'Angers, in tight collaboration with the Phenotic platform of UMR IRHS INRAe, in Angers, France, 2017 - 2020. The use cases chosen for illustration in this thesis. reflects the various collaborations developed on the platform. This included seedling growth [72], weed/crop classification [362, 265], leaf segmentation [363] and apple detection [364]. The list of our publications are listed in the following.

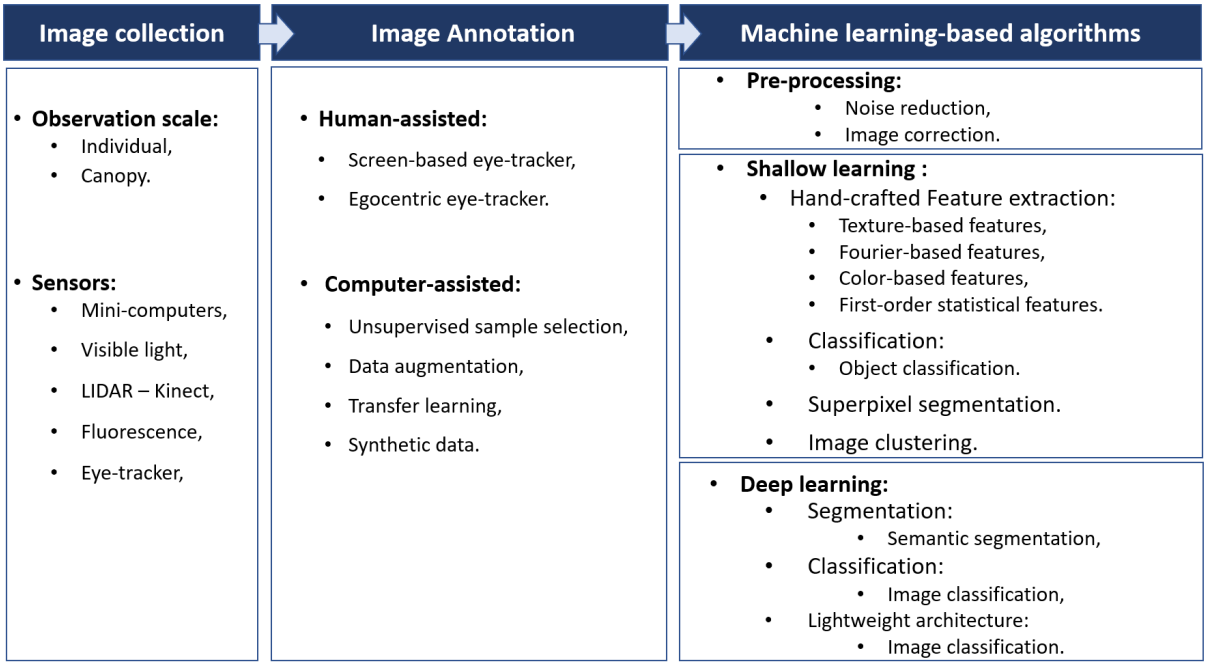


Figure 4.1 – A synthetic view of the spectrum of methods. These methods have been developed/used in this thesis for different plant phenotyping questions, including, monitoring seedling growth in individual and canopy scale, diagnosing plant biotic and abiotic stress, detection of leaf area, fruits and weeds, and more.

Journal Articles

- Salma Samiei, Pejman Rasti, Hervé Daniel, Etienne Belin, Paul Richard, and David Rousseau. "Toward a computer vision perspective on the visual impact of vegetation in symmetries of urban environments". In: *Symmetry* (2018).
- Pejman Rasti, Christian Wolf, Hugo Dorez, Raphael Sablong, Driffa Moussata, Salma Samiei, and David Rousseau. "Machine Learning-Based Classification of the Health State of Mice Colon in Cancer Study from Confocal Laser Endomicroscopy". In: *Scientific Reports.* (Dec. 2019).
- Pejman Rasti, Ali Ahmad, Salma Samiei, Etienne Belin, and David Rousseau. "Supervised Image Classification by Scattering Transform with Application to Weed Detection in Culture Crops of High Density". In: *Remote Sensing.* (2019).
- Salma Samiei, Pejman Rasti, Paul Richard, Gilles Galopin, and David Rousseau. "Toward joint acquisition-annotation of images with egocentric devices for lower-cost machine learning application to apple detection".(2020).
- Salma Samiei, Pejman Rasti, Joseph Ly Vu, Buitink Julia, and David Rousseau. "Deep learning-based detection of seedling development". (2020).

International Conferences

- Natalia Sapoukhina, Salma Samiei, Pejman Rasti, and David Rousseau. "Data Augmentation From RGB to Chlorophyll Fluorescence Imaging Application to Leaf Segmentation of Arabidopsis thaliana From Top View Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR - CVPPP).* Long Beach, CA, 2019.
- Salma Samiei, Ali Ahmad, Pejman Rasti, Etienne Belin, and David Rousseau. "Low-cost image annotation for supervised machine learning. Application to the detection of weeds in dense culture". In: *British Machine Vision Conference (BMVC, CVPPP).* Newcastle, England, 2018 - **Best poster award**.
- Salma Samiei, Pejman Rasti, and David Rousseau. "Low-cost video annotation by voice for supervised machine learning." In: *European Conference on Computer Vision (ECCV, CVPPP).* Glasgow, Scotland, 2020.
- Salma Samiei, Ali Ahmad, Pejman Rasti, and David Rousseau. "New cost and bottleneck in the Era of machine learning-based bioimage analysis". In: *The 3rd NEUBIAS Conference.* Luxembourg, Luxembourg, 2019, **Invited Talk**.

National Conferences

- Salma Samiei, Pejman Rasti, François Chapeau-Blondeau, and David Rousseau. *Cultivons notre jardin avec Fourier. In 27ème Colloque GRETSI sur le Traitement du Signal et des Images.* Lille, France, 2019.
- Salma Samiei, Pejman Rasti, Hervé Daniel, Etienne Belin, Paul Richard, and David Rousseau. *Réalité virtuelle et vision par ordinateur au service de la végétalisation des espaces urbains. In 10ème Rencontres du Végétal.* 2018.

4.2 Perspectives

This work opens new perspectives. Some of them have been highlighted at the end of each contribution detailed in the manuscript. Others could also be pointed here as a further extension of the work presented in the core of this document.

In this thesis, we focused on plant phenotyping in a growth chamber or greenhouse for experiments under relatively well-controlled conditions. While phenotyping in the field is a relatively well-covered topic, a new topic for plant phenotyping is the observation of plants in the urban environment. Understanding the response of plants to stress in such an environment is still to be discovered. Also, plants offers human services in urban ecosystems and imaging may help the study of cities as ecosystems to investigate how to organize vegetation. Annex 1 proposes a first attempt in this direction.

During this thesis we had an opportunity to participate to a study dedicated to biomedical imaging. This included the development of fast image annotation tool for videos. The annotation was speed up with the help of non supervised learning as detailed in Annex 2. This approach could also be tested on the seedling growth imaging of Chapter 2. We currently investigate this perspective.

.

BIBLIOGRAPHY

[1] Yann Chéné et al., « On the use of depth camera for 3D phenotyping of entire plants », *in*: *Computers and Electronics in Agriculture* 82 (2012), pp. 122–127.

[2] Daniel Reynolds et al., « What is cost-efficient phenotyping? Optimizing costs for different scenarios », *in*: *Plant Science* 282.*June 2018* (2019), pp. 14–22.

[3] Massimo Minervini, Hanno Scharr, and Sotirios A Tsaftaris, « Image analysis: the new bottleneck in plant phenotyping [applications corner] », *in*: *IEEE signal processing magazine* 32.*4* (2015), pp. 126–131.

[4] Jose C. Tovar et al., « Raspberry Pi-powered imaging for plant phenotyping », *in*: *Applications in Plant Sciences* 6.*3* (2018).

[5] Lei Li, Qin Zhang, and Danfeng Huang, « A review of imaging techniques for plant phenotyping », *in*: *Sensors (Switzerland)* 14.*11* (2014), pp. 20078–20111.

[6] Massimo Minervini et al., « Phenotiki: an open software and hardware platform for affordable and easy image-based phenotyping of rosette-shaped plants », *in*: *The Plant Journal* 90.*1* (2017), pp. 204–216.

[7] Manuel Vázquez-Arellano et al., « 3-D Imaging Systems for Agricultural Applications—A Review », *in*: *Sensors* 16.*5* (2016), p. 618.

[8] Franck Golbach et al., « Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping », *in*: *Machine Vision and Applications* 27.*5* (2016), pp. 663–680.

[9] Michio Kise, Qin Zhang, and Francisco Rovira-Más, « A Stereovision-based Crop Row Detection Method for Tractor-automated Guidance », *in*: *Biosystems Engineering* 90.*4* (2005), pp. 357–367.

[10] Nan An et al., « Plant high-throughput phenotyping using photogrammetry and imaging techniques to measure leaf length and rosette area », *in*: *Computers and Electronics in Agriculture* 127 (2016), pp. 376–394.

[11] Nan An et al., « Quantifying time-series of leaf morphology using 2D and 3D photogrammetry methods for high-throughput plant phenotyping », *in*: *Computers and Electronics in Agriculture* 135 (2017), pp. 222–232.

[12] Benjamin Franchetti et al., « Vision Based Modeling of Plants Phenotyping in Vertical Farming under Artificial Lighting », *in*: *Sensors* 19.*20* (2019), p. 4378.

[13] Angelika Czedik-Eysenberg et al., « The 'PhenoBox', a flexible, automated, open-source plant phenotyping solution », *in*: *New Phytologist* 219.*2* (2018), pp. 808–823.

[14] Andrei Dobrescu et al., « A "Do-It-Yourself" phenotyping system: measuring growth and morphology throughout the diel cycle in rosette shaped plants », *in*: *Plant Methods* 13.*1* (2017), p. 95.

[15] Karim Panjvani, Anh V Dinh, and Khan A Wahid, « LiDARPheno - A Low-Cost LiDAR-Based 3D Scanning System for Leaf Morphological Trait Extraction. », *in*: *Frontiers in plant science* 10 (2019), p. 147.

[16] He Huang et al., « PCH1 integrates circadian and light-signaling pathways to control photoperiod-responsive growth in Arabidopsis », *in*: *eLife* 5 (2016).

[17] Benoît Valle et al., « PYM: a new, affordable, image-based method using a Raspberry Pi to phenotype plant leaf area in a wide diversity of environments », *in*: *Plant Methods* 13.*1* (2017), p. 98.

[18] Hovári Miklós et al., « IOLT smart pot: An IoT-cloud solution for monitoring plant growth in greenhouses », *in*: *CLOSER - Proceedings of the 9th International Conference on Cloud Computing and Services Science*, SciTePress, 2019, pp. 144–152.

[19] Gustavo A. Pereyra-Irujo et al., « GlyPh: a low-cost platform for phenotyping plant growth and water use », *in*: *Functional Plant Biology* 39.*11* (2012), p. 905.

[20] Wenyi Cao et al., « Quantifying Variation in Soybean Due to Flood Using a Low-Cost 3D Imaging System », *in*: *Sensors* 19.*12* (2019), p. 2682.

[21] Andrew M. Mutka et al., « Quantitative, image-based phenotyping methods provide insight into spatial and temporal dimensions of plant disease », *in*: *Plant Physiology* (2016), pp.00984.2016.

[22] Fu-Ming Lu Ta-Te Lin Wei-Jung Chen, « Integration of a spatial mapping system using GPS and stereo machine vision », *in*: *American Society of Agricultural Engineers (ASAE)*, 2002.

[23] Francisco Rovira-Más, Quanyi Zhang, and John Franklin Reid, « Creation of Three-dimensional Crop Maps based on Aerial Stereoimages », *in*: *Biosystems Engineering* 90.*3* (2005), pp. 251–259.

[24] Harnaik Dhami et al., « Crop Height and Plot Estimation from Unmanned Aerial Vehicles using 3D LiDAR », *in*: (2019), arXiv: `1910.14031`.

[25] Adar Vit and Guy Shani, « Comparing RGB-D Sensors for Close Range Outdoor Agricultural Phenotyping », *in*: *Sensors* 18.*12* (2018), p. 4413.

[26] Francisca López-Granados et al., « An efficient RGB-UAV-based platform for field almond tree phenotyping: 3-D architecture and flowering traits », *in*: *Plant Methods* 15.*1* (2019), p. 160.

[27] Adrian Gracia-Romero et al., « UAV and Ground Image-Based Phenotyping: A Proof of Concept with Durum Wheat », *in*: *Remote Sensing* 11.*10* (2019).

[28] Jaume Casadesús et al., « Using vegetation indices derived from conventional digital cameras as selection criteria for wheat breeding in water-limited environments », *in*: *Annals of Applied Biology* 150.*2* (2007), pp. 227–236.

[29] Jiangui Liu and Elizabeth Pattey, « Retrieval of leaf area index from top-of-canopy digital photography over agricultural crops », *in*: *Agricultural and Forest Meteorology* 150.*11* (2010), pp. 1485–1490.

[30] Hamlyn G. Jones and Robin A. Vaughan, « Remote Sensing of Vegetation: Principles, Techniques, and Applications. », *in*: *The Quarterly Review of Biology* 87.*2* (2012), pp. 165–166.

[31] Kyu-Jong Lee and Byun-Woo Lee, « Estimation of rice growth and nitrogen nutrition status using color digital camera image analysis », *in*: *European Journal of Agronomy* 48 (2013), pp. 57–65.

[32] Philippe Foucher. et al., « Morphological Image Analysis for the Detection of Water Stress in Potted Forsythia », *in*: *Biosystems Engineering* 89.*2* (2004), pp. 131–138.

[33] Wajahat Kazmi et al., « Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison », *in*: *ISPRS Journal of Photogrammetry and Remote Sensing* 88 (2014), pp. 128–146.

[34] Massimo Minervini, Mohammed M. Abdelsamea, and Sotirios A. Tsaftaris, « Image-based plant phenotyping with incremental learning and active contours », *in*: *Ecological Informatics* 23.*September* (2014), pp. 35–48.

[35] Mario Valerio Giuffrida, Massimo Minervini, and Sotirios Tsaftaris, « Learning to Count Leaves in Rosette Plants », *in*: *BMVC- Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, 2015, pp. 1.1–1.13.

[36] Rita Armoniené et al., « Affordable Imaging Lab for Noninvasive Analysis of Biomass and Early Vigour in Cereal Crops », *in*: *BioMed Research International* 2018 (2018), pp. 1–9.

[37] Anja Hartmann et al., « HTPheno: An image analysis pipeline for high-throughput plant phenotyping », *in*: *BMC Bioinformatics* 12.*1* (2011), p. 148.

[38] Noah Fahlgren et al., « A Versatile Phenotyping System and Analytics Platform Reveals Diverse Temporal Responses to Water Availability in Setaria », *in*: *Molecular Plant* 8.*10* (2015), pp. 1520–1535.

[39] Hsien Ming Easlon and Arnold J. Bloom, « Easy Leaf Area: Automated Digital Image Analysis for Rapid and Accurate Measurement of Leaf Area », *in*: *Applications in Plant Sciences* 2.*7* (2014), p. 1400033.

[40] Tino Dornbusch et al., « Measuring the diurnal pattern of leaf hyponasty and growth in Arabidopsis - a novel phenotyping approach using laser scanning », *in*: *Functional Plant Biology* 39.*11* (2012), p. 860.

[41] Andrej A Arsovski et al., « Photomorphogenesis », *in*: *The Arabidopsis Book/American Society of Plant Biologists* 10 (2012).

[42] Babette Dellen, Hanno Scharr, and Carme Torras, « Growth signatures of rosette plants from time-lapse video », *in*: *IEEE/ACM transactions on computational biology and bioinformatics* 12.*6* (2015), pp. 1470–1478.

[43] Sruti Das Choudhury, Ashok Samal, and Tala Awada, « Leveraging image analysis for High-Throughput plant phenotyping », *in*: *Frontiers in plant science* 10 (2019).

[44] Jan F Humplık et al., « Bayesian approach for analysis of time-to-event data in plant biology », *in*: *Plant Methods* 16.*1* (2020), p. 14.

[45] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, « Deep learning », *in*: *nature* 521.*7553* (2015), pp. 436–444.

[46] Andreas Kamilaris and Francesc X. Prenafeta-Boldú, « Deep learning in agriculture: A survey », *in*: *Computers and Electronics in Agriculture* 147.*July 2017* (2018), pp. 70–90.

[47] Yoshihiro Sako et al., « A system for automated seed vigour assessment », *in*: *Seed science and technology* 29.*3* (2001), pp. 625–636.

[48] Alex Hoffmaster et al., « An automated system for vigor testing three-day-old soybean seedlings », *in*: *Seed Science and Technology* 31.*3* (2003), pp. 701–713.

[49] Júlio Marcos-Filho et al., « Assessment of melon seed vigour by an automated computer imaging system compared to traditional procedures », *in*: *Seed Science and Technology* 34.*2* (2006), pp. 485–497.

[50] Julio Marcos Filho, Ana Lúcia Pereira Kikuti, and Liana Baptista de Lima, « Procedures for evaluation of soybean seed vigor, including an automated computer imaging system », *in*: *Revista Brasileira de Sementes* 31.*1* (2009), pp. 102–112.

[51] Ronny V. L. Joosen et al., « germinator: a software package for high-throughput scoring and curve fitting of Arabidopsis seed germination », *in*: *The Plant Journal* 62.*1* (2010), pp. 148–159.

[52] Étienne Belin et al., « Thermography as non invasive functional imaging for monitoring seedling growth », *in*: *Computers and electronics in agriculture* 79.*2* (2011), pp. 236–240.

[53] Landry Benoit et al., « Computer vision under inactinic light for hypocotyl–radicle separation with a generic gravitropism-based criterion », *in*: *Computers and Electronics in Agriculture* 111 (2015), pp. 12–17.

[54] Julio Marcos Filho, « Seed vigor testing: an overview of the past, present and future perspective », *in*: *Scientia Agricola* 72.*4* (2015), pp. 363–374.

[55] Friederike Gnädinger and Urs Schmidhalter, « Digital counts of maize plants by unmanned aerial vehicles (UAVs) », *in*: *Remote sensing* 9.*6* (2017), p. 544.

[56] Pejman Rasti et al., « Low-cost vision machine for high-throughput automated monitoring of heterotrophic seedling growth on wet paper support. », *in*: *BMVC*, 2018, p. 323.

[57] Ruizhi Chen et al., « Monitoring cotton (Gossypium hirsutum L.) germination using ultrahigh-resolution UAS images », *in*: *Precision agriculture* 19.*1* (2018), pp. 161–177.

[58] Biquan Zhao et al., « Rapeseed seedling stand counting and seeding performance evaluation at two early growth stages based on unmanned aerial vehicle imagery », *in*: *Frontiers in plant science* 9 (2018), p. 1362.

[59] Yu Jiang et al., « DeepSeedling: deep convolutional network and Kalman filter for plant seedling detection and counting in the field », *in*: *Plant methods* 15.*1* (2019), p. 141.

[60] Sebastian Kipp et al., « High-throughput phenotyping early plant vigour of winter wheat », *in*: *European Journal of Agronomy* 52 (2014), pp. 271–278.

[61] Sindhuja Sankaran, Lav R Khot, and Arron H Carter, « Field-based crop phenotyping: Multispectral aerial imaging for evaluation of winter wheat emergence and spring stand », *in*: *Computers and Electronics in Agriculture* 118 (2015), pp. 372–379.

[62] Mehedi Hasan et al., « Detection and analysis of wheat spikes using convolutional neural networks », *in*: *Plant Methods* 14.*1* (2018), p. 100.

[63] Pouria Sadeghi-Tehran et al., « Automated method to determine two critical growth stages of wheat: heading and flowering », *in*: *Frontiers in plant science* 8 (2017), p. 252.

[64] Rafael C Gonzalez, Richard E Woods, and Barry R. Masters, *Digital Image Processing, Third Edition*, 2009.

[65] Richard Szeliski, *Computer Vision*, Texts in Computer Science, London: Springer London, 2011.

[66] Diederik Kingma and Jimmy Ba, « Adam: A Method for Stochastic Optimization », *in*: *International Conference on Machine Learning (ICML)*, 2015.

[67] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, « Speech recognition with deep recurrent neural networks », *in*: *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, pp. 6645–6649.

[68] Sepp Hochreiter and Jürgen Schmidhuber, « Long short-term memory », *in*: *Neural computation* 9.*8* (1997), pp. 1735–1780.

[69] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, « On the difficulty of training recurrent neural networks », *in*: *International conference on machine learning*, 2013, pp. 1310–1318.

[70] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, « Sequence to sequence learning with neural networks », *in*: *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[71] Xingjian Shi et al., « Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting », *in*: *Advances in neural information processing systems*, 2015, pp. 68–80, arXiv: 1506.04214.

[72] Salma Samiei et al., « Cultivons notre jardin avec Fourier », *in*: *27ème Colloque GRETSI sur le Traitement du Signal et des Images, Lille, France.* 2019.

[73] Mads Dyrmann, Søren Skovsen, and Rasmus Nyholm Jørgensen, « Hierarchical multi-label Classification of Plant Images using Convolutional Neural Network », *in*: *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, 2019.

[74] Jonghoon Jin et al., « Tracking with deep neural networks », *in*: *2013 47th Annual Conference on Information Sciences and Systems (CISS)*, IEEE, 2013, pp. 1–5.

[75] Ralph Bours et al., « OSCILLATOR: A system for analysis of diurnal leaf growth using infrared photography combined with wavelet transformation », *in*: *Plant Methods* 8.*1* (2012), pp. 1–12.

[76] Robertson McClung, « Plant circadian rhythms », *in*: *The Plant Cell* 18.*4* (2006), pp. 792–803.

[77] Jean Baptiste Joseph baron Fourier, *Théorie analytique de la chaleur*, F. Didot publisher., 1822.

[78] Antoine Cornuéjols and Laurent Miclet, *Apprentissage artificiel: concepts et algorithmes*, Editions Eyrolles, 2011.

[79] David. G. Stork, « Character and document research in the Open Mind Initiative », *in*: *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*, 1999, pp. 1–12.

[80] Jeff Howe, *Crowdsourcing: Why the power of the crowd is driving the future of business*, Random House, 2008.

[81] Eric Hand, « Citizen science: People power », *in*: *Nature News* 466.*7307* (2010), pp. 685–687.

[82] Marisa Ponti et al., « Getting it Right or Being Top Rank: Games in Citizen Science », *in*: *Citizen Science: Theory and Practice* 3.*1* (2018), pp. 1–12.

[83] Edith Law and Luis von Ahn, *Human Computation*, 1st, Morgan & Claypool Publishers, 2011.

[84] Nurulhasanah Mazlan, Sharifah Sakinah Syed Ahmad, and Massila Kamalrudin, « A Crowdsourcing Approach for Volunteering System Using Delphi Method », *in*: *Innovative Computing, Optimization and Its Applications: Modelling and Simulations*, Cham: Springer International Publishing, 2018, pp. 237–253.

[85] Luis von Ahn and Laura Dabbish, « Labeling Images with a Computer Game », *in*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, Vienna, Austria: ACM, 2004, pp. 319–326.

[86] Luis Von Ahn, Ruoran Liu, and Manuel Blum, « Peekaboom: A game for locating objects in images », *in*: *Conference on Human Factors in Computing Systems - Proceedings*, vol. 1, 2006, pp. 55–64.

[87] Seth Cooper et al., « Predicting protein structures with a multiplayer online game. », *in*: *Nature* 466.*1* (2010), pp. 756–760.

[88] Srinivas C Turaga et al., « Space-time wiring specificity supports direction selectivity in the retina. », *in*: *Nature* 509.*7500* (2014), pp. 331–6.

[89] Robert Simpson, Kevin R Page, and David De Roure, « Zooniverse: observing the world's largest citizen science platform », *in*: *Proceedings of the 23rd international conference on world wide web*, ACM, 2014, pp. 1049–1054.

[90] Benjamin Yao, Xiong Yang, and Song-Chun Zhu, « Introduction to a Large-Scale General Purpose Ground Truth Database: Methodology, Annotation Tool and Benchmarks », *in*: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2007.

[91] Carsten Saathoff et al., « M-OntoMat-Annotizer: Linking Ontologies with Multimedia Low-Level Features for Automatic Image Annotation », *in*: *Poster & Demo Session, ESWC 2006*, 2006.

[92] Christian Halaschek-Wiener et al., « Photostuff - an image annotation tool for the semantic web », *in*: *4th International Semantic Web Conference Poster Paper*, 2005.

[93] Laura Hollink et al., « Adding Spatial Semantics to Image Annotations », *in*: *International Workshop on Knowledge Markup and Semantic Annotation*, 2004.

[94] *Flicker-*, `https://www.flickr.com/`.

[95] Timo Volkmer, Jonathan Simon Richardson Smith, and Apostol Natsev, « A web-based system for collaborative annotation of large image and video collections: an evaluation and user study », *in*: *ACM Multimedia*, 2005.

[96] Jeroen Steggink and Cees G.M. Snoek, « Adding semantics to image-region annotations with the Name-It-Game », *in*: *Multimedia Systems* 17.*5* (2011), pp. 367–378.

[97] Chris J. Lintott et al., « Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey », *in*: *Monthly Notices of the Royal Astronomical Society* 389 (2008), pp. 1179–1189.

[98] Veronika Cheplygina et al., « Early Experiences with Crowdsourcing Airway Annotations in Chest CT », *in*: *CoRR* abs/1706.02055 (2017), arXiv: `1706.02055`.

[99] Amaia Salvador et al., « Crowdsourced Object Segmentation with a Game », *in*: *Proceedings of the 2Nd ACM International Workshop on Crowdsourcing for Multimedia*, CrowdMM '13, Barcelona, Spain: ACM, 2013, pp. 15–20.

[100] Duarte Gonçalves et al., « Tag around: A 3D Gesture Game for Image Annotation », *in*: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, Yokohama, Japan: Association for Computing Machinery, 2008, pp. 259–262.

[101] Stefan Thaler et al., « SeaFish: A Game for Collaborative and Visual Image Annotation and Interlinking », *in*: *The Semanic Web: Research and Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 466–470.

[102] Chien-Ju Ho et al., « KissKissBan: A Competitive Human Computation Game for Image Annotation », *in*: *SIGKDD Explor. Newsl.* 12.*1* (2010), pp. 21–24.

[103] Lasantha Seneviratne and Ebroul Izquierdo, « Image annotation through gaming (TAG4FUN) », *in*: *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–6.

[104] *EteRNA - Annotation Platform.* `http://eterna.stanford.edu`, 2016.

[105] Debra A. Fischer et al., « Planet Hunters: the first two planet candidates identified by the public using the Kepler public archive data », *in*: *Monthly Notices of the Royal Astronomical Society* 419.*4* (2011), pp. 2900–2911.

[106] Sam Mavandadi et al., « Distributed Medical Image Analysis and Diagnosis through Crowd-Sourced Games: A Malaria Case Study », *in*: *PLOS ONE* 7.*5* (2012), pp. 1–8.

[107] Sam Mavandadi et al., « BioGames: A Platform for Crowd-Sourced Biomedical Image Analysis and Telediagnosis », *in*: *Games for health* 1 (2012), pp. 373–376.

[108] Miguel Angel Luengo-Oroz, Asier Arranz, and John Frean, « Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears », *in*: *Journal of medical Internet research* 14.*6* (2012), e167–e167.

[109] Luis Von Ahn, Mihir Kedia, and Manuel Blum, « Verbosity: a game for collecting common-sense facts », *in*: *In Proceedings of ACM CHI - Conference on Human Factors in Computing Systems, volume 1 of Games*, ACM Press, 2006, pp. 75–78.

[110] Rui Jesus et al., « Playing games as a way to improve automatic image annotation », *in*: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2008, pp. 1–8.

[111] Ville H Tuulos, Jürgen Scheible, and Heli Nyholm, « Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup BT - Pervasive Computing », *in*: *International Conference on Pervasive Computing*, Springer Berlin Heidelberg, 2007, pp. 37–54.

[112] Adela Barriuso and Antonio Torralba, « Notes on image annotation », *in*: *CoRR* abs/1210.3448 (2012), arXiv: `1210.3448`.

[113] Severin Hacker and Luis von Ahn, « Matchin: Eliciting User Preferences with an Online Game », *in*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, Boston, MA, USA: ACM, 2009, pp. 1207–1216.

[114] Michiel Hildebrand et al., « Waisda?: video labeling game », *in*: *ACM Multimedia*, 2013.

[115] Bryan C. Russell et al., « LabelMe: A Database and Web-Based Tool for Image Annotation », *in*: *International Journal of Computer Vision* 77 (2005), pp. 157–173.

[116] Bryan C. Russell et al., « LabelMe: A database and web-based tool for image annotation », *in*: *International Journal of Computer Vision* 77.*1-3* (2008), pp. 157–173.

[117] Alexander Sorokin and David Forsyth, « Utility data annotation with Amazon Mechanical Turk », *in*: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops c* (2008), pp. 1–8.

[118] Vaggelis Mourelatos, Manolis Tzagarakis, and Efthalia Dimara, « A review of online crowdsourcing platforms », *in*: *South-Eastern Europe Journal of Economics* 14 (2016), pp. 59–74.

[119] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis, « Running experiments on Amazon mechanical turk », *in*: *Judgment and Decision Making* 5.*5* (2010), pp. 411–419.

[120] Adriana Kovashka et al., « Crowdsourcing in Computer Vision », *in*: *Foundations and Trends® in Computer Graphics and Vision* 10.*3* (2016), pp. 177–243.

[121] Lena Maier-Hein et al., « Can Masses of Non-Experts Train Highly Accurate Image Classifiers? », *in*: *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, Cham: Springer International Publishing, 2014, pp. 438–445.

[122] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani, « Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria », *in*: *Proceedings of the NAACL HLT- Workshop on Active Learning for Natural Language Processing*, Boulder, Colorado: Association for Computational Linguistics, 2009, pp. 27–35.

[123] Lena. Maier-Hein et al., « Crowd-Algorithm Collaboration for Large-Scale Endoscopic Image Annotation with Confidence », *in*: *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, Springer International Publishing, 2016, pp. 616–623.

[124] Eric Heim et al., « Clickstream analysis for crowd-based object segmentation with confidence », *in*: *CoRR* abs/1611.08527 (2016).

[125] Silas Nyboe Ørting et al., « A Survey of Crowdsourcing in Medical Image Analysis », *in*: *CoRR* abs/1902.09159 (2019), arXiv: `1902.09159`.

[126] Carsten Eickhoff, « Crowd-powered Experts: Helping Surgeons Interpret Breast Cancer Images », *in*: *Proceedings of the First International Workshop on Gamification for Information Retrieval*, GamifIR '14, Amsterdam, The Netherlands: ACM, 2014, pp. 53–56.

[127] Shadi Albarqouni et al., « AggNet: Deep Learning From Crowds for Mitosis Detection in Breast Cancer Histology Images », *in*: *IEEE Transactions on Medical Imaging* 35.*5* (2016), pp. 1313–1321.

[128] Benjamin M. Good and Andrew I. Su, « Crowdsourcing for bioinformatics », *in*: *Bioinformatics* 29.*16* (2013), pp. 1925–1933.

[129] Mohamed Amgad et al., « Structured crowdsourcing enables convolutional segmentation of histology images », *in*: *Bioinformatics* (2019).

[130] Humayun Irshad et al., « Crowdsourcing scoring of immunohistochemistry images: Evaluating Performance of the Crowd and an Automated Computational Method », *in*: *Scientific Reports* 7.*February* (2017), pp. 1–10.

[131] Humayun Irshad et al., « Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: Evaluating experts, automated methods, and the crowd », *in*: *Pacific Symposium on Biocomputing* (2015), pp. 294–305.

[132] Julien Minet et al., « Crowdsourcing for agricultural applications: A review of uses and opportunities for a farmsourcing approach », *in*: *Computers and Electronics in Agriculture* 142.*November* (2017), pp. 126–138.

[133] Naihui Zhou et al., « Crowdsourcing image analysis for plant phenomics to generate ground truth data for machine learning », *in*: *PLoS Computational Biology* 14.*7* (2018), pp. 1–20.

[134] Stefanie Nowak and Stefan Rüger, « How Reliable Are Annotations via Crowdsourcing: A Study About Inter-annotator Agreement for Multi-label Image Annotation », *in*: *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, Philadelphia, Pennsylvania, USA: ACM, 2010, pp. 557–566.

[135] Osamuyimen Stewart, David Lubensky, and Juan M. Huerta, « Crowdsourcing Participation Inequality: A SCOUT Model for the Enterprise Domain », *in*: *Proceedings of the ACM, Workshop on Human Computation*, HCOMP '10, Washington DC, 2010, pp. 30–33.

[136] Raphaël Marée et al., « Cytomine: An Open-Source Software For Collaborative Analysis Of Whole-Slide Images », *in*: *Diagnostic Pathology* 1.*8* (2016).

[137] Virginia Gulick, « Clickworkers Interactive : Progress on a JPEG2000-Streaming Annotation Interface », *in*: *Lunar and Planetary Science Conference-LPSC*, 2014.

[138] *Colabeler*, `http://www.colabeler.com`.

[139] Massimo Minervini, Mario Valerio Giuffrida, and Sotirios Tsaftaris, « An interactive tool for semi-automated leaf annotation », *in*: *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, BMVA Press, 2015, pp. 6.1–6.13.

[140] *samasource- Annotation Platform.* `https://www.samasource.com/`.

[141] *Spare5- Annotation Platform.* `https://app.spare5.com/fives`.

[142] *Cloudfactory*, `https://www.cloudfactory.com/`.

[143] *Infolks- Annotation Platform.* `https://infolks.info/`.

[144] Zsolt Palotai et al., « LabelMovie: Semi-supervised Machine Annotation Tool with Quality Assurance and Crowd-sourcing Options for Videos », *in*: *Proceedings - International Workshop on Content-Based Multimedia Indexing*, 2014.

[145] *Supervisely- Annotation Platform.* `https://supervise.ly/`.

[146] Abhishek Dutta and Andrew Zisserman, « The VGG Image Annotator (VIA) », *in*: *CoRR* abs/1904.10699 (2019), arXiv: `1904.10699`.

[147] *VGG Image Annotator- Annotation Platform.* `http://www.robots.ox.ac.uk/~vgg/software/via/`.

[148] *labelbox- Annotation Platform.* `https://labelbox.com/`.

[149] *Labelme- Annotation Platform.* `https://github.com/wkentaro/labelme`.

[150] Carl Vondrick, Deva Ramanan, and Donald Patterson, « Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces », *in*: *Computer Vision – ECCV*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 610–623.

[151] *rectlabel- Annotation Platform.* `https://rectlabel.com/`.

[152] Xuebin Qin et al., « ByLabel: A Boundary Based Semi-Automatic Image Annotation Tool », *in*: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1804–1813.

[153] Philipe A. Dias et al., « FreeLabel: A Publicly Available Annotation Tool based on Freehand Traces », *in*: *CoRR* abs/1902.06806 (2019), arXiv: 1902.06806.

[154] *prodigy- Annotation Platform.* https://prodi.gy/.

[155] *Trainingdata- Annotation Platform.* https://www.trainingdata.io/.

[156] *Pixel Annotation Tool- Annotation Platform.* https://github.com/abreheret/PixelAnnotationTool.

[157] *Cogito- Annotation Platform.* https://www.cogitotech.com/.

[158] *Aispotters- Annotation Platform.* https://www.aispotters.com/.

[159] *InfoSearch- Annotation Platform.* https://www.infosearchbpo.com/.

[160] *Hive AI - Annotation Platform.* https://thehive.ai/.

[161] *Computer Vision Annotation Tool (CVAT)- Annotation Platform.* https://github.com/opencv/cvat.

[162] *Meetbunch- Annotation Platform.* https://www.meetbunch.com/.

[163] *Scale- Annotation Platform.* https://scale.com.

[164] *Oclavi- Annotation Platform.* https://oclavi.com/.

[165] *Microwork- Annotation Platform.* https://microwork.io/.

[166] Imanol Luengo et al., « SuRVoS: Super-Region Volume Segmentation workbench », *in*: *Journal of Structural Biology* 198.*1* (2017), pp. 43–53.

[167] Christoph Sommer et al., « Ilastik: Interactive learning and segmentation toolkit », *in*: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 230–233.

[168] *Fast Image Data Annotation Tool (FIAT)- Annotation Platform.* ://github.com/christopher5106/FastAnnotationTool.

[169] *Taqadam- Annotation Platform.* https://taqadam.io/.

[170] *Human in the loop - Annotation Platform.* https://humansintheloop.org/.

[171] *Playment - Annotation Platform.* https://playment.io.

[172] *Workaround- Annotation Platform.* https://workaround.online/.

[173] *Diffgram- Annotation Platform.* https://diffgram.com/.

[174] *Linked AI- Annotation Platform.* https://platform.linkedai.co/.

[175] *Haizaha- Annotation Platform.* https://www.haizaha.com/.

[176] Niklas Fiedler, Marc Bestmann, and Norman Hendrich, « ImageTagger: An Open Source Online Platform for Collaborative Image Labeling », *in*: *RoboCup- Robot World Cup XXII*, Springer, 2018.

[177] *Visual Object Tagging Tool (VoTT)- Annotation Platform.* https://github.com/microsoft/VoTT.

[178] *Anno-Mage*, https://github.com/virajmavani/semi-auto-image-annotation-tool.

[179] Stephan Saalfeld et al., « CATMAID: collaborative annotation toolkit for massive amounts of image data », *in*: *Bioinformatics* 25.*15* (2009), pp. 1984–1986.

[180] Hennie Brugman et al., « EUDICO, Annotation and Exploitation of Multi Media Corpora over the Internet », *in*: *Measuring Behavior 3rd International Conference on Methods and Techniques in Behavioral Research*, Nijmegen, The Netherlands, 2000.

[181] *Alp- Annotation platform.* https://alpslabel.wordpress.com/2017/01/26/alt/.

[182] Pongsate Tangseng, Zhipeng Wu, and Kota Yamaguchi, « Looking at Outfit to Parse Clothing », *in*: *CoRR* abs/1703.01386 (2017), arXiv: 1703.01386.

[183] *Label Object and Save Time (LOST)- Annotation Platform.* https://lost.readthedocs.io/.

[184] *Sloth- Annotation Platform.* https://cvhci.anthropomatik.kit.edu/~baeuml/projects/a-universal-labeling-tool-for-computer-vision-sloth/.

[185] *Claysciences- Annotation Platform.* https://www.claysciences.com/.

[186] Julius Schöning, Patrick Faion, and Gunther Heidemann, « Semi-automatic Ground Truth Annotation in Videos: An InteractiveTool for Polygon-based Object Annotation and Segmentation », *in*: *International Conference on Knowledge Capture (K-CAP)*, ACM, New York, 2015, 17:1–17:4.

[187] Julius Schöning, Patrick Faion, and Gunther Heidemann, « Pixel-wise Ground Truth Annotation in Videos - An Semi-automaticApproach for Pixel-wise and Semantic Object Annotation », *in*: *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, SciTePress, 2016, pp. 690–697.

[188] *hasty AI- Annotation Platform.* `https://hasty.ai/`.

[189] Daniel Rubin et al., « ePAD: An Image Annotation and Analysis Platform for Quantitative Imaging », *in*: *Tomography (Ann Arbor, Mich.)* 5 (2019), pp. 170–183.

[190] *Coco Annotator- Annotation Platform.* `https://github.com/jsbroks/coco-annotator/`.

[191] Nuria Teixido et al., « Hierarchical segmentation-based software for cover classification analyses of seabed images (Seascape) », *in*: *Marine Ecology Progress Series* 431 (2011), pp. 45–53.

[192] *handl AI- Annotation Platform.* `https://handl.ai/`.

[193] *Ultimate Labeling- Annotation Platform.* `https://github.com/alexandre01/UltimateLabeling`.

[194] *Open Labeler- Annotation Platform.* `https://github.com/kinhong/OpenLabeler`.

[195] *Alturos Image Annotation- Annotation Platform.* `https://github.com/AlturosDestinations/`.

[196] *OpenLabeling- Annotation Platform.* `https://github.com/Cartucho/OpenLabeling`.

[197] *Yolo mark- Annotation Platform.* `https://github.com/AlexeyAB/Yolo_mark`.

[198] *Deep Label- Annotation Platform.* `https://github.com/jveitchmichaelis/deeplabel`.

[199] *Pixie- Annotation Platform.* `https://github.com/buni-rock/Pixie`.

[200] *KNOSSOS- Annotation Platform.* `https://knossos.app/`.

[201] Jiří Borovec, Jan Kybic, and Rodrigo Nava, « Detection and Localization of Drosophila Egg Chambers in Microscopy Images », *in*: *Machine Learning in Medical Imaging: 8th International Workshop, MLMI*, Springer International Publishing, 2017, pp. 19–26.

[202] Akihiro Sugimoto Jiří Borovec Jan Kybic, « Region growing using superpixels with learned shape prior », *in*: *Journal of Electronic Imaging* 26.*6* (2017).

[203] *LCFinder- Annotation Platform.* `https://github.com/lc-soft/LC-Finder`.

[204] *Understand- Annotation Platform.* `https://understand.ai/`.

[205] Philip Dawid et al., « Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm », *in*: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.*1* (1979), pp. 20–28.

[206] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis, « Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers », *in*: *International Conference on Knowledge Discovery and Data Mining (14th ACM SIGKDD)*, KDD '08, Las Vegas, Nevada, USA: Association for Computing Machinery, 2008, pp. 614–622.

[207] Padhraic Smyth, Usama Fayyad, and Michael Burl, « Inferring ground truth from subjective labelling of venus images », *in*: *Advances in Neural Information Processing Systems 7* (1995), pp. 1085–1092.

[208] Merrielle Spain and Pietro Perona, « Measuring and Predicting Object Importance », *in*: *International Journal of Computer Vision* 91.*1* (2011), pp. 59–76.

[209] Luis von Ahn et al., « reCAPTCHA: Human-Based Character Recognition via Web Security Measures », *in*: *Science* 321.*5895* (2008), pp. 1465–1468.

[210] Jacob Whitehill et al., « Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise », *in*: *Advances in Neural Information Processing Systems* 22.*1* (2009), pp. 1–9.

[211] Vikas C. Raykar et al., « Supervised Learning from Multiple Experts: Whom to Trust when Everyone Lies a Bit », *in*: *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, Montreal, Quebec, Canada: ACM, 2009, pp. 889–896.

[212] Sudheendra Vijayanarasimhan and Kristen Grauman, « What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations », *in*: vol. IEEE, June, 2009, pp. 2262–2269.

[213] Peter Welinder and Pietro Perona, « Online crowdsourcing: Rating annotators and obtaining cost-effective labels », *in*: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 25–32.

[214] Yuji Roh, Geon Heo, and Steven Euijong Whang, « A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective », *in*: *CoRR* abs/1811.03402 (2018), arXiv: 1811.03402.

[215] Josef Kittler Leila Shafarenko Maria Petrou, « Automatic watershed segmentation of randomly textured color images », *in*: *IEEE Transactions on Image Processing* 6.*11* (1997), pp. 1530–1544.

[216] Kevis-Kokitsi Maninis et al., « Deep Extreme Cut: From Extreme Points to Object Segmentation », *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 616–625.

[217] Radhakrishna Achanta et al., « SLIC superpixels compared to state-of-the-art superpixel methods », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.*11* (2012), pp. 2274–2281.

[218] Gerard Sychay, Edward Y. Chang, and King-Shy Goh, « Effective image annotation via active learning », *in*: *IEEE International Conference on Multimedia and Expo*, IEEE, pp. 209–212.

[219] Kaiming He et al., « Mask R-CNN », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, 2017, pp. 2980–2988, arXiv: arXiv:1703.06870v3.

[220] Sinno Jialin Pan and Qiang Yang, « A Survey on Transfer Learning », *in*: *IEEE Transactions on Knowledge and Data Engineering* 22.*10* (2010), pp. 1345–1359.

[221] Jason Yosinski et al., « How transferable are features in deep neural networks? », *in*: *CoRR* abs/1411.1792 (2014).

[222] Jia Deng et al., « ImageNet: A large-scale hierarchical image database », *in*: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[223] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé, « Using Deep Learning for Image-Based Plant Disease Detection », *in*: *Frontiers in Plant Science* 7 (2016).

[224] Jordan R. Ubbens and Ian Stavness, « Corrigendum: Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks », *in*: *Frontiers in Plant Science* 8.*July* (2018).

[225] Aniruddha Tapas, « Transfer learning for image classification and plant phenotyping », *in*: *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)* 5.*11* (2016), pp. 2664–2669.

[226] Clément Douarre et al., « Transfer Learning from Synthetic Data Applied to Soil–Root Segmentation in X-Ray Tomography Images », *in*: *Journal of Imaging* 4.*5* (2018).

[227] Ekin Dogus Cubuk et al., « AutoAugment: Learning Augmentation Policies from Data », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[228] Yezi Zhu et al., « Data Augmentation using Conditional Generative Adversarial Networks for Leaf Counting in Arabidopsis Plants », *in*: *The British Machine Vision Conference (BMVC)- Computer Vision Problems in Plant Phenotyping (CVPPP)*, 2018, pp. 1–11.

[229] Dmitry Kuznichov et al., « Data Augmentation for Leaf Segmentation and Counting Tasks in Rosette Plants », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019), pp. 2580–2589.

[230] Przemyslaw Prusinkiewicz, « Graphical Applications of L-Systems », *in*: *Proceedings on Graphics Interface '86/Vision Interface '86*, Vancouver, British Columbia, Canada: Canadian Information Processing Society, 1986, pp. 247–253.

[231] Alexander Buslaev et al., « Albumentations: Fast and Flexible Image Augmentations », *in*: *Information* 11.*2* (Feb. 2020), p. 125.

[232] Mario Valerio Giuffrida, Hanno Scharr, and Sotirios A. Tsaftaris, « ARIGAN: Synthetic arabidopsis plants using generative adversarial network », *in*: *Proceedings - IEEE International Conference on Computer Vision Workshops, ICCVW*, IEEE, 2017, pp. 2064–2071, arXiv: 1709.00938.

[233] Landry Benoit et al., « Simulation of image acquisition in machine vision dedicated to seedling elongation to validate image processing root segmentation algorithms », *in*: *Computers and Electronics in Agriculture* 104 (2014), pp. 84–92.

[234] Aditya Jain et al., « A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students », *in*: *International Journal of Applied and Basic Medical Research* 5.*2* (2015), p. 124.

[235] Tina Walber, Ansgar Scherp, and Steffen Staab, « Can You See It? Two Novel Eye-Tracking-Based Measures for Assigning Tags to Image Regions », *in*: *International Conference in Advances in Multimedia Modeling*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 36–46.

[236] Kiwon Yun et al., « Studying relationships between human gaze, description, and computer vision », *in*: *Computer vision and pattern recognition (cvpr), 2013 ieee conference on*, IEEE, 2013, pp. 739–746.

[237] Timo Ojala, Matti Pietikainen, and Topi Maenpaa, « Multiresolution gray-scale and rotation invariant texture classification with local binary patterns », *in*: *IEEE Transactions on pattern analysis and machine intelligence* 24.*7* (2002), pp. 971–987.

[238] Robert M Haralick, « Statistical and structural approaches to texture », *in*: *Proceedings of the IEEE*, vol. 67, 5, IEEE, 1979, pp. 786–804.

[239] Ishella S Fogel and Dalit Sagi, « Gabor filters as texture discriminator », *in*: *Biological Cybernetics* 61.*2* (1989).

[240] Corinna Cortes and Vladimir Vapnik, « Support-vector networks », *in*: *Machine Learning* 20.*3* (1995), pp. 273–297.

[241] Robert M Haralick, Karthikeyan Shanmugam, and Dinstein Its'Hak, « Textural features for image classification », *in*: *IEEE Transactions on systems, man, and cybernetics 6* (1973), pp. 610–621.

[242] Scott A Shearer et al., « Plant identification using color co-occurrence matrices », *in*: *Transactions of the ASAE* 33.*6* (1990), pp. 1237–1244.

[243] Thomas Burks, Scott A. Shearer, and Fred A Payne, « Classification of weed species using color texture features and discriminant analysis », *in*: *Transactions of the ASAE* 43.*2* (2000), p. 441.

[244] Young Ki Chang et al., « Development of color co-occurrence matrix based machine vision algorithms for wild blueberry fields », *in*: *Applied engineering in agriculture* 28.*3* (2012), pp. 315–323.

[245] Alireza Fathi, Ali Farhadi, and James M. Rehg, « Understanding egocentric activities », *in*: *Proceedings of the IEEE International Conference on Computer Vision* (2011), pp. 407–414.

[246] Aiden R. Doherty et al., « Passively recognising human activities through lifelogging », *in*: *Computers in Human Behavior* 27.*5* (2011), pp. 1948–1958.

[247] Hamed Pirsiavash and Deva Ramanan, « Detecting activities of daily living in first-person camera views », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2847–2854.

[248] Zheng Lu and Kristen Grauman, « Story-driven summarization for egocentric video », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 2714–2721.

[249] Alireza Fathi, Xiaofeng Ren, and James M. Rehg, « Learning to recognize objects in egocentric activities », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011), pp. 3281–3288.

[250] Luca Erculiani, Fausto Giunchiglia, and Andrea Passerini, « Continual egocentric object recognition », *in*: *Computer Vision and Pattern Recognition* (2019), arXiv: 1912.05029.

[251] Andrew J. Davison et al., « MonoSLAM: Real-time single camera SLAM », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.*6* (2007), pp. 1052–1067.

[252] Alejandro Rituerto, « Modeling the environment with egocentric vision systems », *in*: *Electronic Letters on Computer Vision and Image Analysis* 14.*3* (2015), pp. 49–51.

[253] Stefano Alletto et al., « Understanding social relationships in egocentric vision », *in*: *Pattern Recognition* 48.*12* (2015), pp. 4082–4096.

[254] Alejandro Betancourt et al., « The Evolution of First Person Vision Methods: A Survey », *in*: *IEEE Transactions on Circuits and Systems for Video Technology* 25.*5* (2015), pp. 744–760, arXiv: 1409.1484.

[255] Kuang Yu Liu, Shih Chung Hsu, and Chung Lin Huang, « First-person-vision-based driver assistance system », *in*: *ICALIP 2014 - 2014 International Conference on Audio, Language and Image Processing*, IEEE, 2015, pp. 239–244.

[256] Walterio W. Mayol et al., « Applying active vision and SLAM to wearables », *in*: *Springer Tracts in Advanced Robotics*, vol. 15, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 325–334.

[257] Svebor Karaman et al., « Human daily activities indexing in videos from wearable cameras for monitoring of patients with dementia diseases », *in*: *Proceedings - International Conference on Pattern Recognition*, IEEE, 2010, pp. 4113–4116.

[258] Aiden R. Doherty et al., « Wearable cameras in health: The state of the art and future possibilities », *in*: *American Journal of Preventive Medicine* 44.*3* (2013), pp. 320–323.

[259] Yin Li, Alireza Fathi, and James M. Rehg, « Learning to predict gaze in egocentric video », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2013, pp. 3216–3223.

[260] Cheng Li and Kris M. Kitani, « Pixel-level hand detection in ego-centric videos », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, pp. 3570–3577.

[261] Sven Bambach et al., « Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions », *in*: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, IEEE, 2015, pp. 1949–1957.

[262] Minghuang Ma, Haoqi Fan, and Kris M. Kitani, « Going Deeper into First-Person Activity Recognition », *in*: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, IEEE, 2016, pp. 1894–1903, arXiv: 1605.03688.

[263] Benjamin W. Tatler, Roland J. Baddeley, and Iain D. Gilchrist, « Visual correlates of fixation selection: Effects of scale and time », *in*: *Vision Research* 45.5 (2005), pp. 643–659.

[264] Tina Walber, « Making use of eye tracking information in image collection creation and region annotation », *in*: *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, New York, New York, USA: ACM Press, 2012, pp. 1405–1408.

[265] Salma Samiei et al., « Low-cost image annotation for supervised machine learning. Application to the detection of weeds in dense culture. », *in*: *British Machine Vision Conference (BMVC) , Computer Vision Problems in Plant Phenotyping (CVPPP)*, Newcastle, UK: BMVA Press, 2018.

[266] Alfredo Lucas et al., « Image Annotation by Eye Tracking: Accuracy and Precision of Centerlines of Obstructed Small-Bowel Segments Placed Using Eye Trackers », *in*: *Journal of Digital Imaging* 32.5 (2019), pp. 855–864.

[267] Edward A. Parrish, « Pictorial Pattern Recognition Applied To Fruit Harvesting. », *in*: *Transactions of the American Society of Agricultural Engineers* 20.5 (1977), pp. 822–827.

[268] Antoine Grand D'esnon et al., « Magali: A self-propelled robot to pick apples », *in*: *American Society of Agricultural Engineering Paper* 46 (1987), pp. 353–358.

[269] Dale Whittaker et al., « Fruit Location in a Partially Occluded Image. », *in*: *Transactions of the American Society of Agricultural Engineers* 30.*3* (1987), pp. 591–596.

[270] David C Slaughter et al., « Color vision in robotic fruit harvesting », *in*: *Transactions of the ASAE* 30.*4* (1987), pp. 1144–1148.

[271] Peter Sites and Michael Delwiche, « Computer Vision To Locate Fruit on a Tree. », *in*: *Transactions of the American Society of Agricultural Engineers* 31.*1* (1988), pp. 257–263, 272.

[272] G.A. Rabatel, « A vision system for Magali, the fruit picking robot », *in*: *International Conference on Agricultural Engineering (Ageng'88) Paris, France.* 1988, pp. 1–6.

[273] L. Kassay, « Hungarian robotic apple harvester. », *in*: *Proceedings of the ASAE Annual Meeting Papers, St. Joseph; MI, USA. 4–6* (1992), pp. 1–14.

[274] Ramón Ceres et al., « Agribot : A Robot for Aided Fruit Harvesting », *in*: *Industrial Robot* 25.*5* (1998), pp. 337–46.

[275] Antonio Ramón Jiménez, Ramón Ceres, and Jose Pons, « A survey of computer vision methods for locating fruit on trees », *in*: *Transactions of the American Society of Agricultural Engineers* 43.*6* (2000), pp. 1911–1920.

[276] Rong Zhou et al., « Using colour features of cv. 'Gala' apple fruits in an orchard in image processing to predict yield », *in*: *Precision Agriculture* 13.*5* (2012), pp. 568–580.

[277] Yu Song et al., « Automatic fruit recognition and counting from multiple images », *in*: *Biosystems Engineering* 118.*1* (2014), pp. 203–215.

[278] Inkyu Sa et al., « Deepfruits: A fruit detection system using deep neural networks », *in*: *Sensors* 16.*8* (2016), p. 1222.

[279] Frans Boogaard., Kamiel S.A.H. Rongen, and Gert W. Kootstra, « Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging », *in*: *Biosystems Engineering* 192 (2020), pp. 117–132.

[280] Qi Wang et al., « Automated Crop Yield Estimation for Apple Orchards », *in*: *Experimental Robotics. Springer Tracts in Advanced Robotics* (2013), pp. 745–758.

[281] Calvin Hung et al., « A feature learning based approach for automated fruit yield estimation », *in*: *Springer Tracts in Advanced Robotics*, vol. 105, Cham: Springer International Publishing, 2015, pp. 485–498.

[282] Suchet Bargoti and James Underwood, « Image classification with orchard metadata », *in*: *Proceedings - IEEE International Conference on Robotics and Automation* (2016), pp. 5164–5170.

[283] Suchet Bargoti and James P. Underwood, « Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards », *in*: *Journal of Field Robotics* 34.*6* (2017), pp. 1039–1060, arXiv: 1610.08120.

[284] Pieter Blignaut, « Fixation identification: The optimum threshold for a dispersion algorithm », *in*: *Attention, Perception, & Psychophysics* 71.*4* (2009), pp. 881–895.

[285] Keith Rayner, « Eye movements in reading and information processing: 20 years of research. », *in*: *Psychological Bulletin* 124.*3* (1998), pp. 372–422.

[286] Robert Jacob, « What You Look at is What You Get: Eye Movement-Based Interaction Techniques », *in*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI 90, Seattle, Washington, USA: Association for Computing Machinery, 1990, pp. 11–18.

[287] David E. Irwin, *Eye Movements and Visual Cognition: Scene Perception and Reading*, New York, NY: Springer New York, 1992, pp. 146–165.

[288] Robert Jacob, « Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces », *in*: *Advances in human-computer interaction*, 1993.

[289] Dario Salvucci and Joseph Goldberg, « Identifying Fixations and Saccades in Eye-Tracking Protocols », *in*: *Proceedings of the 2000 Symposium on Eye Tracking Research Applications*, ETRA '00, Palm Beach Gardens, Florida, USA: Association for Computing Machinery, 2000, pp. 71–78.

[290] Barry R. Manor and Evian Gordon, « Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks », *in*: *Journal of Neuroscience Methods* 128.*1-2* (2003), pp. 85–93.

[291] Andrew Duchowski, *Eye Tracking Methodology*, Springer London., 2007.

[292] Frederick Shic, Brian Scassellati, and Katarzyna Chawarska, « The incomplete fixation measure », *in*: *Proceedings of the symposium on Eye tracking research & applications - ETRA '08*, New York, New York, USA: ACM Press, 2008, p. 111.

[293] Oleg Spakov and Darius Miniotas, « Application of clustering algorithms in eye gaze visualization. », *in*: *Information Technology and Control* 36 (2015).

[294] Sergey Zagoruyko and Nikos Komodakis, « Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer », *in*: *5th International Conference on Learning Representations, ICLR - Conference Track Proceedings*, vol. abs/1612.0, 2019, arXiv: 1612.03928.

[295] Omri Safren et al., « Detection of green apples in hyperspectral images of apple-tree foliage using machine vision », *in*: *Transactions of the ASABE* 50.*6* (2007), pp. 2303–2313.

[296] Jordi Gené-Mola et al., « KFuji RGB-DS database: Fuji apple multi-modal images for fruit detection with color, depth and range-corrected IR data », *in*: *Data in Brief* 25 (2019), p. 104289.

[297] Nicolai Hani, Pravakar Roy, and Volkan Isler, « MinneApple: A Benchmark Dataset for Apple Detection and Segmentation », *in*: *IEEE Robotics and Automation Letters* 5.*2* (2020), pp. 852–858, arXiv: 1909.06441.

[298] Hanwen Kang and Chao Chen, « Fruit detection and segmentation for apple harvesting using visual sensor in orchards », *in*: *Sensors (Switzerland)* 19.*20* (2019), p. 4599.

[299] Pravakar Roy et al., « Vision-based preharvest yield mapping for apple orchards », *in*: *Computers and Electronics in Agriculture* 164 (2019), p. 104897, arXiv: 1808.04336.

[300] Xiaoyang Liu et al., « The recognition of apple fruits in plastic bags based on block classification », *in*: *Precision Agriculture* 19.*4* (2018), pp. 735–749.

[301] Xiaoyang Liu et al., « A Detection Method for Apple Fruits Based on Color and Shape Features », *in*: *IEEE Access* 7 (2019), pp. 67923–67933.

[302] John A. Hartigan, « Algorithm AS 136: A K-Means Clustering Algorithm », *in*: *Applied Statistics* 28.*1* (1979), p. 100.

[303] *SMI Eye Tracking- Annotation Platform.* https://gazeintelligence.com/smi-software-download.

[304] Radhakrishna Achanta et al., « Salient region detection and segmentation », *in*: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5008 LNCS, ICVS 2008, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 66–75.

[305] Laurent Itti, Christof Koch, and Ernst Niebur, « A model of saliency-based visual attention for rapid scene analysis », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.*11* (1998), pp. 1254–1259.

[306] Radhakrishna Achanta et al., « Frequency-tuned salient region detection », *in*: *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 1597–1604.

[307] Jean-Michel Pape and Christian Klukas, « 3-D histogram-based segmentation and leaf detection for rosette plants », *in*: *European Conference on Computer Vision*, Springer, 2014, pp. 61–74.

[308] Kyle Simek and Kobus Barnard, « Gaussian process shape models for bayesian segmentation of plant leaves », *in*: *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP), pages* (2015), pp. 4–1.

[309] Jean-Michel Pape and Christian Klukas, « Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images », *in*: *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)* (2015), pp. 1–12.

[310] Sotirios A. Tsaftaris, Massimo Minervini, and Hanno Scharr, « Machine Learning for Plant Phenotyping Needs Image Processing », *in*: *Trends in Plant Science* 21.*12* (2016), pp. 989–991.

[311] Daniel Ward, Peyman Moghadam, and Nicolas Hudson, « Deep Leaf Segmentation Using Synthetic Data », *in*: *Proceedings of the British Machine Vision Conference (BMVC) Workshop on Computer Vision Problems in Plant Phenotyping (CVPPP)*, 2018, arXiv: `1807.10931`.

[312] Warren L Butler, « Energy distribution in the photochemical apparatus of photosynthesis », *in*: *Annual Review of Plant Physiology* 29.*1* (1978), pp. 345–378.

[313] Stephen Alexander Rolfe and Julie Diane Scholes, « Chlorophyll fluorescence imaging of plant-pathogen interactions », *in*: *Protoplasma* 247.*3-4* (2010), pp. 163–175.

[314] Céline Rousseau et al., « High throughput quantitative phenotyping of plant resistance using chlorophyll fluorescence image analysis », *in*: *Plant methods* 9.*1* (2013), p. 17.

[315] Justine Bresson et al., « Quantifying spatial heterogeneity of chlorophyll fluorescence during plant growth and in response to water stress », *in*: *Plant Methods* 11.*1* (2015), p. 23.

[316] Andrew M Mutka et al., « Quantitative, image-based phenotyping methods provide insight into spatial and temporal dimensions of plant disease », *in*: *Plant physiology* 172.*2* (2016), pp. 650–660.

[317] Ruud Barth et al., « Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset », *in*: *Computers and Electronics in Agriculture* 144 (2018), pp. 284–296.

[318] Maurilio Di Cicco et al., « Automatic model based dataset generation for fast and accurate crop and weeds detection », *in*: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 5188–5195.

[319] Yuta Hiasa et al., « Cross-modality image synthesis from unpaired data using CycleGAN », *in*: *International Workshop on Simulation and Synthesis in Medical Imaging*, Springer, 2018, pp. 31–41.

[320] Chawin Ounkomol et al., « Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy », *in*: *Nature methods* 15.*11* (2018), p. 917.

[321] Massimo Minervini et al., « Finely-grained annotated datasets for image-based plant phenotyping », *in*: *Pattern Recognition Letters* 81 (2016), pp. 80–89.

[322] Daniel Ward and Peyman Moghadam, « Synthetic arabidopsis dataset. », *in*: *Commonwealth Scientific and Industrial Research Organisation (CSIRO). Data Collection.* (2018).

[323] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, « U-net: Convolutional networks for biomedical image segmentation », *in*: *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[324] Kaiming He et al., « Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification », *in*: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, USA: IEEE Computer Society, 2015, pp. 1026–1034.

[325] *Segmentation Models- Annotation Platform.* `https://github.com/qubvel/segmentation_models`.

[326] Serge Beucher, « Use of watersheds in contour detection », *in*: *Proceedings of the International Workshop on Image Processing*, CCETT, 1979.

[327] Serge Beucher, « The watershed transformation applied to image segmentation », *in*: *Scanning Microscopy International* 6.*1* (1991), pp. 299–314.

[328] Carole H Sudre, Wenqi Li, and M. Jorge Cardoso Tom Vercauteren Sébastien Ourselin, « Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations », *in*: *International Workshop on Deep Learning in Medical Image Analysis International Workshop on Multimodal Learning for Clinical Decision Support*, 2017.

[329] Andrew Janowczyk and Anant Madabhushi, « Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases », *in*: *Journal of pathology informatics* 7 (2016).

[330] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang, « A survey of transfer learning », *in*: *Journal of Big Data* 3.*1* (2016), p. 9.

[331] Joan Bruna and Stéphane Mallat, « Invariant scattering convolution networks », *in*: *IEEE transaction on pattern analysis and machine intelligence* (2013), pp. 1872–1886.

[332] Shervin Minaee, AmirAli Abdolrashidi, and Yao Wang, « Iris recognition using scattering transform and textural features », *in*: *Signal Processing and Signal Processing Education Workshop (SP/SPE), IEEE*, IEEE, 2015, pp. 37–42.

[333] Mathieu Lagrange et al., « Classification of rainfall radar images using the scattering transform », *in*: *Journal of Hydrology* 556 (2018), pp. 972–979.

[334] Bran Hongwei Li, Jianguo Zhang, and Wei-Shi Zheng, « HEp-2 cells staining patterns classification via wavelet scattering network and random forest », *in*: *Pattern Recognition (ACPR), 3rd IAPR Asian Conference on*, IEEE, 2015, pp. 406–410.

[335] Alain Rakotomamonjy et al., « Scattering features for lung cancer detection in fibered confocal fluorescence microscopy images », *in*: *Artificial intelligence in medicine* 61.*2* (2014), pp. 105–118.

[336] Xudong Yang et al., « Automatic 3d facial expression recognition using geometric scattering representation », *in*: *Automatic Face and Gesture Recognition (FG), 11th IEEE International Conference and Workshops on*, vol. 1, IEEE, 2015, pp. 1–6.

[337] Jorge Torres-Sánchez et al., « Configuration and specifications of an unmanned aerial vehicle (UAV) for early site specific weed management », *in*: *PloS one* 8.*3* (2013), e58210.

[338] José M Peña et al., « Quantifying efficacy and limits of unmanned aerial vehicle (UAV) technology for weed seedling detection as affected by sensor resolution », *in*: *Sensors* 15.*3* (2015), pp. 5609–5626.

[339] César Fernández-Quintanilla et al., « Is the current state of the art of weed monitoring suitable for site-specific weed management in arable crops? », *in*: *Weed Research* (2018).

[340] Adel Bakhshipour and Abdolabbas Jafari, « Evaluation of support vector machine and artificial neural networks in weed detection using shape features », *in*: *Computers and Electronics in Agriculture* 145 (2018), pp. 153–160.

[341] Philipp Lottes et al., « UAV-based crop and weed classification for smart farming », *in*: *Robotics and Automation (ICRA), IEEE International Conference on*, IEEE, 2017, pp. 3024–3031.

[342] Andres Milioto, Philipp Lottes, and Cyrill Stachniss, « Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs », *in*: *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2229–2235.

[343] Matthew J. Aitkenhead et al., « Weed and crop discrimination using image analysis and artificial intelligence methods », *in*: *Computers and electronics in Agriculture* 39.*3* (2003), pp. 157–171.

[344] John A. Marchant and Christine M. Onyango, « Comparison of a Bayesian classifier with a multilayer feed-forward neural network using the example of plant/weed/soil discrimination », *in*: *Computers and Electronics in Agriculture* 39.*1* (2003), pp. 3–22.

[345] Pandi Prema and D Murugan, « A novel angular texture pattern (ATP) extraction method for crop and weed discrimination using curvelet transformation », *in*: *ELCVIA Electronic Letters on Computer Vision and Image Analysis* 15.*1* (2016), pp. 27–59.

[346] Ali Ahmad et al., « An Image Processing Method Based on Features Selection for Crop Plants and Weeds Discrimination Using RGB Images », *in*: *International Conference on Image and Signal Processing*, Springer, 2018, pp. 3–10.

[347] Sebastian Haug et al., « Plant classification system for crop/weed discrimination without segmentation », *in*: *Applications of Computer Vision (WACV), IEEE Winter Conference on*, IEEE, 2014, pp. 1142–1149.

[348] Adel Bakhshipour et al., « Weed segmentation using texture features extracted from wavelet sub-images », *in*: *Biosystems Engineering* 157 (2017), pp. 1–12.

[349] Jérémie Bossu et al., « Wavelet transform to discriminate between crop and weed in perspective agronomic images », *in*: *computers and electronics in agriculture* 65.*1* (2009), pp. 133–143.

[350] Ian Goodfellow et al., *Deep learning*, vol. 1, MIT press Cambridge, 2016.

[351] Martın Abadi et al., « TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems », *in*: *CoRR* abs/1603.04467 (2016), arXiv: 1603.04467.

[352] Jacob Manning et al., « Machine-Learning Space Applications on SmallSat Platforms with TensorFlow », *in*: *32nd Annual AIAA/USU, Conference on Small Satellites*, 2018.

[353] Andrew G. Howard et al., « MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications », *in*: *CoRR* abs/1704.04861 (2017), arXiv: 1704.04861.

[354] Mark Sandler et al., « Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation », *in*: *CoRR* abs/1801.04381 (2018), arXiv: 1801.04381.

[355] Nicolas Courty et al., « Optimal transport for domain adaptation », *in*: *IEEE transactions on pattern analysis and machine intelligence* 39.*9* (2017), pp. 1853–1865.

[356] David C. Slaughter, D.Ken Giles, and Daniel Downey, « Autonomous robotic weed control systems: A review », *in*: *Computers and Electronics in Agriculture* 61.*1* (2008), pp. 63–78.

[357] Sulaiman Fadlallah and Khaled Goher, « A review of weed detection and control robots: a world without weeds », *in*: *Advances in Cooperative Robotics*, World Scientific, 2017, pp. 233–240.

[358] Ralph Brown and Scott Noble, « Site-specific weed management: sensing requirements—what do we need to see? », *in*: *Weed Science* 53.*2* (2005), pp. 252–258.

[359] Hanno Scharr et al., « Annotated image datasets of rosette plants », *in*: *Technical Report No. FZJ-2014-03837* (2014), pp. 1–16.

[360] Jordan Ubbens et al., « The use of plant models in deep learning: An application to leaf counting in rosette plants », *in*: *Plant Methods* 14.*1* (2018), p. 6.

[361] Pejman Rasti et al., « Machine Learning-Based Classification of the Health State of Mice Colon in Cancer Study from Confocal Laser Endomicroscopy », *in*: *Scientific Reports.* (2019).

[362] Pejman Rasti et al., « Supervised Image Classification by Scattering Transform with Application to Weed Detection in Culture Crops of High Density », *in*: *Remote Sensing* 11.*3* (2019), p. 249.

[363] Natalia Sapoukhina et al., « Data Augmentation From RGB to Chlorophyll Fluorescence Imaging Application to Leaf Segmentation of Arabidopsis thaliana From Top View Images », *in*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR - CVPPP)*, Long Beach, CA, 2019.

[364] Salma Samiei et al., « New cost and bottleneck in the Era of machine learning-based bioimage analysis », *in*: *The 3rd NEUBIAS Conference*, Luxembourg, Luxembourg, 2019.

[365] Salma Samiei et al., « Réalité virtuelle et vision par ordinateur au service de la végétalisation des espaces urbains. », *in*: *In 10ème Rencontres du Végétal*, 2018.

[366] Salma Samiei et al., « Toward joint acquisition-annotation of images with egocentric devices for lower cost machine learning application to apple detection », *in*: *MDPI- Sensors* (2020).

[367] Salma Samiei et al., « Deep learning-based detection of seedling development », *in*: *Plant method.* (2020).

[368] Salma Samiei et al., « Toward a Computer Vision Perspective on the Visual Impact of Vegetation in Symmetries of Urban Environments », *in*: *Symmetry* 10.*12* (2018), p. 666.

Chapter 5

Annex A

MDPI

Article

Toward a Computer Vision Perspective on the Visual Impact of Vegetation in Symmetries of Urban Environments

Salma SAMIEI[1,2], Pejman RASTI[1,2], Hervé DANIEL[3], Etienne BELIN[1,2], Paul RICHARD[1], and David ROUSSEAU[1,2]

[1] LARIS, Université d'Angers, 62 avenue Notre-Dame-du-Lac, 49000, Angers, France
[2] UMR INRA IRHS, 42 Rue Georges Morel, 49070, Beaucouzé, France
[3] UMR BAGAP, INRA, Agrocampus Ouest, ESA, 2 rue André-Le-Nôtre, 49045, Angers, France
* Correspondence: david.rousseau@univ-angers.fr

Version May 14, 2020 submitted to Journal Not Specified

Abstract: Rapid urbanization is a worldwide critical environmental challenge. With this urban migration soaring, we need to live far more efficiently than we currently do by incorporating the natural world in new and innovative ways. There are a lot of researches on ecological, architectural or aesthetic points of view to address this issue. We present a novel approach to assess the visual impact of vegetation in urban street pedestrian view with the assistance of computer vision metrics. We statistically evaluate the correlations of the amount of vegetation with objective computer vision traits such as Fourier domain, color histogram, and estimated depth from monocular view. We show that increasing vegetation in urban street views breaks the orthogonal symmetries of urban blocks, enriches the color space with fractal-like symmetries and decreases the cues of projective geometry in depth. These uncovered statistical facts are applied to predict the requested amount of vegetation to make urban street views appear like natural images. Interestingly, these amounts are found in accordance with the ecosystemic approach for urban planning. Also, the study opens new questions for the understanding of the link between geometry and depth perception.

Keywords: Natural image statistics, Urban views, Fourier, RGB, fractals, Depth map, Symmetry, urban greenery, projective geometry

1. Introduction

At present, more than half of the world's population is estimated to live in the cities. Urban green spaces (UGS) [1,2] are an important factor of urban streetscape which provides aesthetic, economic, environmental, social, and health benefits to urban residents. Accordingly, societal benefits supplied by UGS to city dwellers are vital to maintain and increase urban citizen's quality of life. The study of the impact of UGS on public health [3–7], to manage the urban ecosystem [8,9] or to assess the aesthetic quality of UGS [10] can benefit from various computer vision based approaches. This includes computer vision to acquire the semantic information of every single pixel of an urban space [11–15] or analyzing the visual impact of vegetation in urban environments [16–18] from top view images in birds or satellite viewpoint.

In this article, we apply computer vision techniques in the urban landscape from the viewpoint of a pedestrian in an urban street. Urban street views are highly geometrical and symmetrical environments with orthogonal and parallel lines radically different from what is found in a natural environment where structures following any orientations are more likely to occur. Also, from a color perspective, urban street views often offer few hue values due to their limited mineral content. This is very different

from the richness of color found in nature. Characterizing the quantitative impact of vegetation on these visual symmetries would enable to assess the question of how much vegetation should be included in an urban street view to make it look more natural than man-made. We provide a computer vision quantification of the impact of vegetation in urban street views by determining their statistical properties.

Obviously, a very large set of potential descriptors could be used for this application. Instead of an exhaustive benchmark of existing computer vision tools of the literature, or an automatic selection of such tools with machine learning approaches, we proceed with an analytical approach where we select simple descriptors which have been successfully applied in the literature to identify statistical invariant features in pure man-made or purely natural scenes. We explore, for the first time to our knowledge, how this small set of historical descriptors behave in presence of various amount of vegetation. We also provide new experiments and dataset specially designed for this work.

As related works, one can recall that understanding the statistical properties of the natural environment is an important problem in computer vision [19–21]. Although explored for decades, characterization of statistical invariance and symmetries in natural images continues to progress. For instance, as recently reviewed in [20], this progress has been obtained by considering first and higher statistics of the luminance and color distributions of natural images, local orientation in images or statistical cues in the natural visual environment that is available to compute disparity. Other approaches have also contributed to the characterization of natural images by considering different categories of them. This includes gazing at natural scenes with new sensors (range camera [22,23], thermal camera [24], polarized light [25] ...) or in various environments of interest for humans (underwater [26], man versus natural environment [27,28]). We adopt this approach of characterizing natural images by considering specific categories. We focus on urban street views with various amount of vegetation. These scenes are therefore important as already underlined for urban planning but also for vision understanding because they constitute an intermediate category between pure man-made scenes and purely natural scenes in which the visual system has evolved originally.

The two main novelties of the paper appear along the following lines : (i) None of the existing strategies for urban greenery has proposed so far a computer vision perspective at a pedestrian level. Computer vision was already used to analyze the statistics of man-made versus wild images but (ii) we propose the first application of these techniques for man-made block world with various amount of vegetation. The article is organized along the following structure. In the second section, we explore how vegetation in urban street view can be quantified with computer vision tools in the Fourier domain. In the following section, we investigate another aspect of the visual impact of urban vegetation in the color domain. The fourth section explains the last aspect of the visual impact of urban vegetation on depth. In conclusion, the application of these results is discussed in terms of urban planning, and we mention new questions now opened for further investigation by this work.

2. Impact of vegetation in the Fourier domain

Different categories of natural environments present different orientations. For instance buildings, for stability reasons, will provide orthogonal edges while plants [29], because they seek for maximum light, are more likely to propose edges in all directions. This has been demonstrated in [28] to statistically translate into a characteristic signature in the power Fourier spectrum of natural images with anisotropy patterns in the Fourier domain of urban views and isotropic patterns in the natural landscape. In this section, we reproduce the experiment of [28] with urban street views including a various amount of vegetation between pure man-made and purely natural scenes.

To this purpose we consider the cityscapes dataset [30] which includes 25,000 RGB annotated street pedestrian view images from 50 cities mainly located in Germany. For experiments in this article, 5,000 images from fine annotations category are used. The annotation includes the labeling of vegetation. It is therefore straightforward to associate a percentage of vegetation with each image of

this dataset. Three examples of this dataset are shown in Fig. 1 with the RGB images, the annotated images and the percentage of vegetation which is evaluated through annotated images.

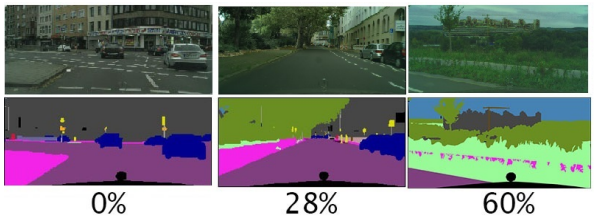


Figure 1. Three examples of images taken from the dataset considered in this study [30]. The first row shows the RGB images. The second row is the corresponding annotated images. The percentage value gives the percentage of vegetation measured from the annotated images.

Figure 2 illustrates spectral signature computed by Fourier transform for the same three images of the dataset as in Fig. 1. As visible in Fig. 2 and similarly to what was found in [28] the presence of plants tends to reduce the anisotropy between the spectral energy in horizontal and vertical frequencies.

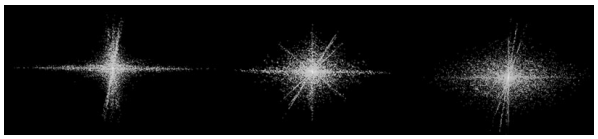


Figure 2. Three examples of the modulus of the Fourier transform of the RGB images of Fig. 1 from left to right with 0%, 28% and 60% of vegetation respectively.

Figure 3 illustrates the full pipeline followed for investigating the impact of vegetation in the Fourier domain. Due to the underexposed aspect of the RGB images in this dataset, CLAHE [31] algorithm is used for adjusting the contrast of the image and intensity equalization. Then, the RGB images are converted into gray levels $L = 0.299 \times R + 0.587 \times G + 0.114 \times B$ and transferred from spatial domain to frequency domain through 2-D Fourier transform. The modulus of this Fourier transform is thresholded, as in [28], in order to keep 70% of the energy. To measure the vertical-horizontal anisotropy of this binarized spectrum the following ratio of orientation is computed as

$$R = \frac{D1 + D2}{\sqrt{H^2 + V^2}}, \tag{1}$$

where, as shown in Fig. 4, D1, and D2 represent the diagonal and antidiagonal size of the spectral signature, H indicates the horizontal size of spectral signature and V represents the vertical size of it.

This anisotropy ratio is then plotted as a function of the percentage of vegetation. Examples of three cities from the dataset are given in Fig. 5 where a linear trend clearly appears with decreasing anisotropy ratio as a function of the percentage of vegetation in the image. The average slope on the whole dataset [30] is weak. However, this trend is systematically found and statistically valid for all cities as demonstrated in table 1.

Experimental results of this Fourier approach demonstrate, as one could intuitively expect from [28], that the presence of vegetation tends to break the horizontal-vertical symmetry in the images as quantified by the anisotropy ratio defined in this section.

We now come to propose a possible application of this statistical result for urban planning. We considered the pure concrete-based images and pure natural images (in the wild) of [28] to determine the expected range of the anisotropy ratio. Natural images were found on average with an anisotropy ratio of 0.84 and pure concrete of 0.74. An intermediate value in this range could constitute a transition limit between an environment perceived as natural and an environment perceived as pure man-made in the Fourier domain. To test this hypothesis, we considered the middle value of this anisotropy ratio between natural and pure concrete environment. This average value (0.79), pointed in green in Fig.

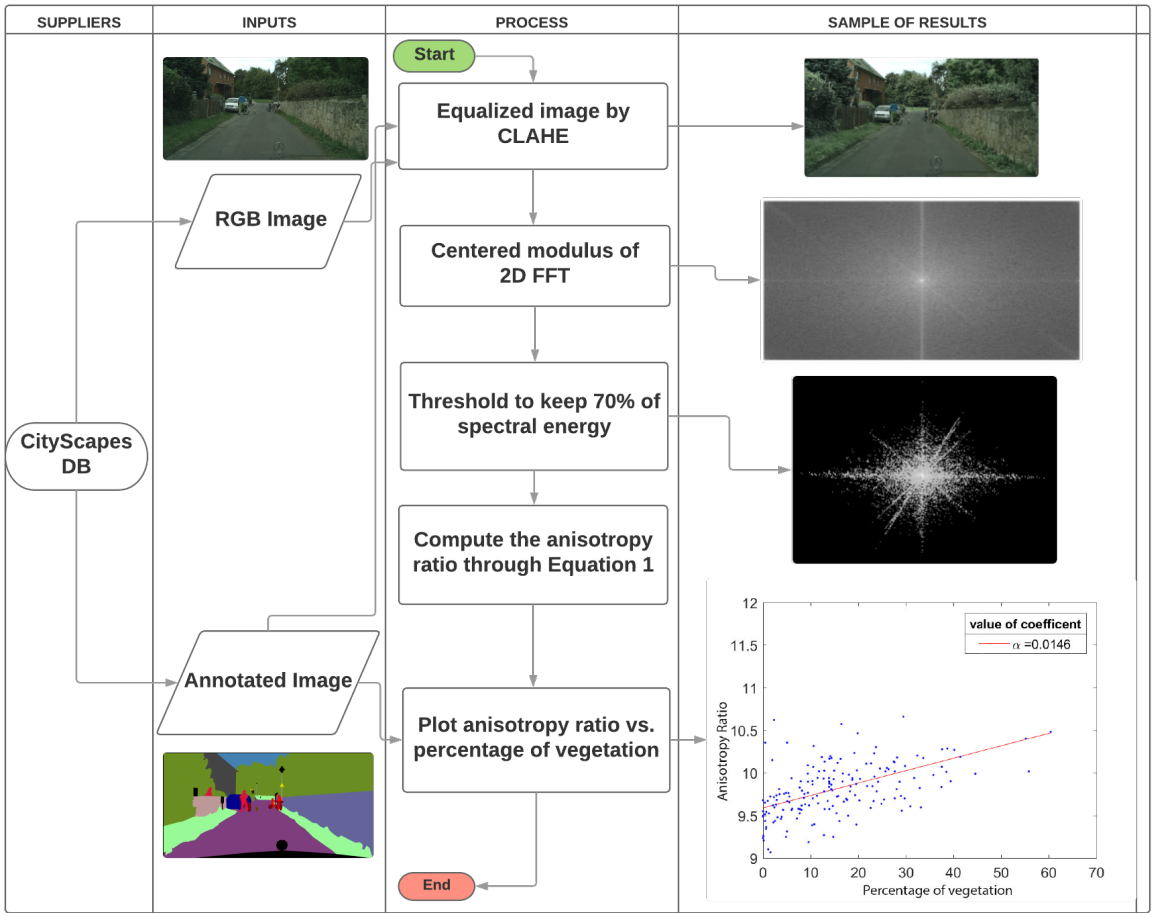


Figure 3. Image processing pipeline for the study of the impact of vegetation in the Fourier domain. CityScapes DB [30] corresponds to an available dataset as illustrated in Fig. 1.

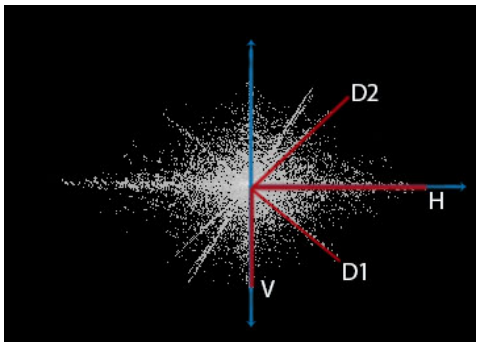


Figure 4. Elements computed in the anisotropy ratio of Eq. (1) applied on the modulus of the Fourier transform.

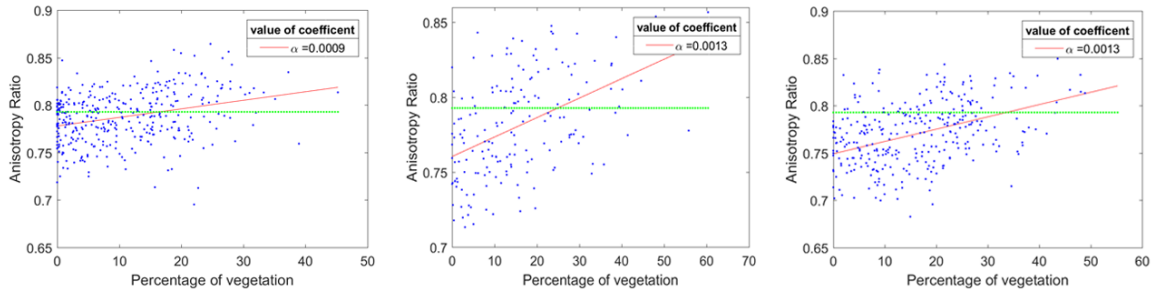


Figure 5. Anisotropy ratio as a function of the percentage of the vegetation of three examples of cities taken from the dataset considered in this study [30]. The left column is Strasbourg city. The column in the middle is for Aachen. The right column is Bremen city. The red line is the linear fit of the data. The green line is the reference 0.79 corresponding to the average anisotropy value between the pure man-made and pure natural environment.

5, crosses the linear fit computed for each city and provides an associated percentage of vegetation. The average value of the requested percentage of vegetation to reach this anisotropy ratio is not found to trivially be 50% but rather 28% with a standard deviation of 8.32%. The requested percentage of vegetation in cities is also debated in other scientific fields. Percentage of vegetation below which fragmentation of urban ecosystems has consequences on the diversity and viability of these ecosystems have been highlighted in [32–34] for instance. Interestingly, it appears that the threshold around which processes are favored or not are found to be between 20 and 30% of vegetation, i.e. in a similar range as the one found here with our computer vision approach. However, it is to be noticed that the decrease of anisotropy ratio statistically recorded could also be obtained without any vegetation by simply using non-orthogonal building architectures promoting, for example, curved shapes with edges in all orientations.

3. Impact of vegetation in the color domain

Another aspect of the visual impact of vegetation on the images is explored in this section. In a pure concrete urban environment, it is likely that the color embedded in the color histogram of images will be limited to some blue in the sky, grey-black on the ground and a small number of colors correlated to the mineral content used for walls of the building in the image (generally not green). Adding vegetation in an urban concrete environment is therefore expected to enrich the color histogram. The statistics of the color histogram of natural images have been studied in [35–37] where scale invariant symmetries were observed in the organization of the color in the RGB 3D histogram. We reproduce similar experiments with the dataset of the previous section [30] to investigate the evolution of the RGB 3D color histogram as a function of the amount of vegetation in the image.

Table 1. The slope of the anisotropy ratio as a function of the percentage of vegetation and associated P-value.

City	Slope	P-value
Aachen	1.3 E-3	8.10 E-11
Bochum	1.4 E-3	1.73 E-07
Bremen	1.3 E-3	8.06 E-17
Cologne	1.1 E-3	1.87 E-06
Darmstadt	1.4 E-3	9.72 E-06
Dusseldorf	1.4 E-3	8.67 E-17
Hamburg	1.0 E-3	4.44 E-08
Hanover	1.4 E-3	4.58 E-12
Jena	1.0 E-3	1.40 E-05
Krefeld	1.5 E-3	5.82 E-06
Monchengladbach	1.5 E-3	1.22 E-07
Strasbourg	0.9 E-3	1.92 E-08
Stuttgart	0.8 E-3	3.35 E-05
Tubingen	0.5 E-3	2.37 E-03
Ulm	1.3 E-3	7.71 E-07
Weimar	1.2 E-3	2.38 E-10
Zurich	0.8 E-3	3.76 E-05

Figure 6 shows three examples of the 3D color histogram for the same images of Fig. 1. As visible in Fig. 6, the presence of plants tends to increase the complexity of the 3D color histogram.

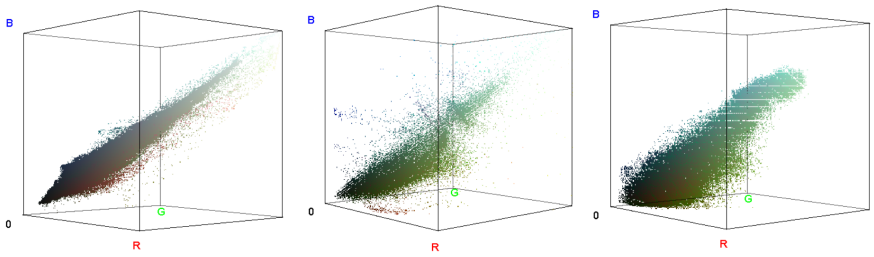


Figure 6. Three examples of the 3D color histogram of the RGB images in Fig. 1. From left to right with 0%, 28% and 60% of vegetation respectively. The presence of plants tends to increase the richness of the 3D color histogram.

Figure 7 illustrates the full pipeline followed for investigating the impact of vegetation on the 3D color histogram. A box-counting procedure is applied as follows. The colorimetric cube is successively covered with boxes of side a and volume a^3, with varying a. For each box of size a, one computes the number $N(a)$ of boxes which are needed to cover the support of the 3D histogram, i.e. to cover all the cells of the colorimetric cube which are occupied by pixels of the image. As observed in [35–37] the evolution of the number of counts $N(a)$ as a function of the color scale on a log-log scale is well approximated by straight lines with slope $-$D, over a significant range of colorimetric scales a. This scale invariant symmetry is associated with a fractal behavior with fractal dimension D. Then for all images in each city, the slope values D are plotted as a function of the percentage of vegetation. Examples for three cities of the dataset are given in Fig. 8 where a linear systematic trend clearly appears over the whole set of cities with increasing fractal dimension D as a function of the percentage of vegetation in the image. Table 2 gives the slope of the fractal dimension and associated p-value for all cities in CityScape dataset.

Experimental results of this approach in the color domain demonstrate that the presence of vegetation tends to enrich the complexity of the 3D color histogram. This enrichment is in the direction of higher dimension scale invariant symmetries as quantified by the box-counting fractal dimension defined in this section.

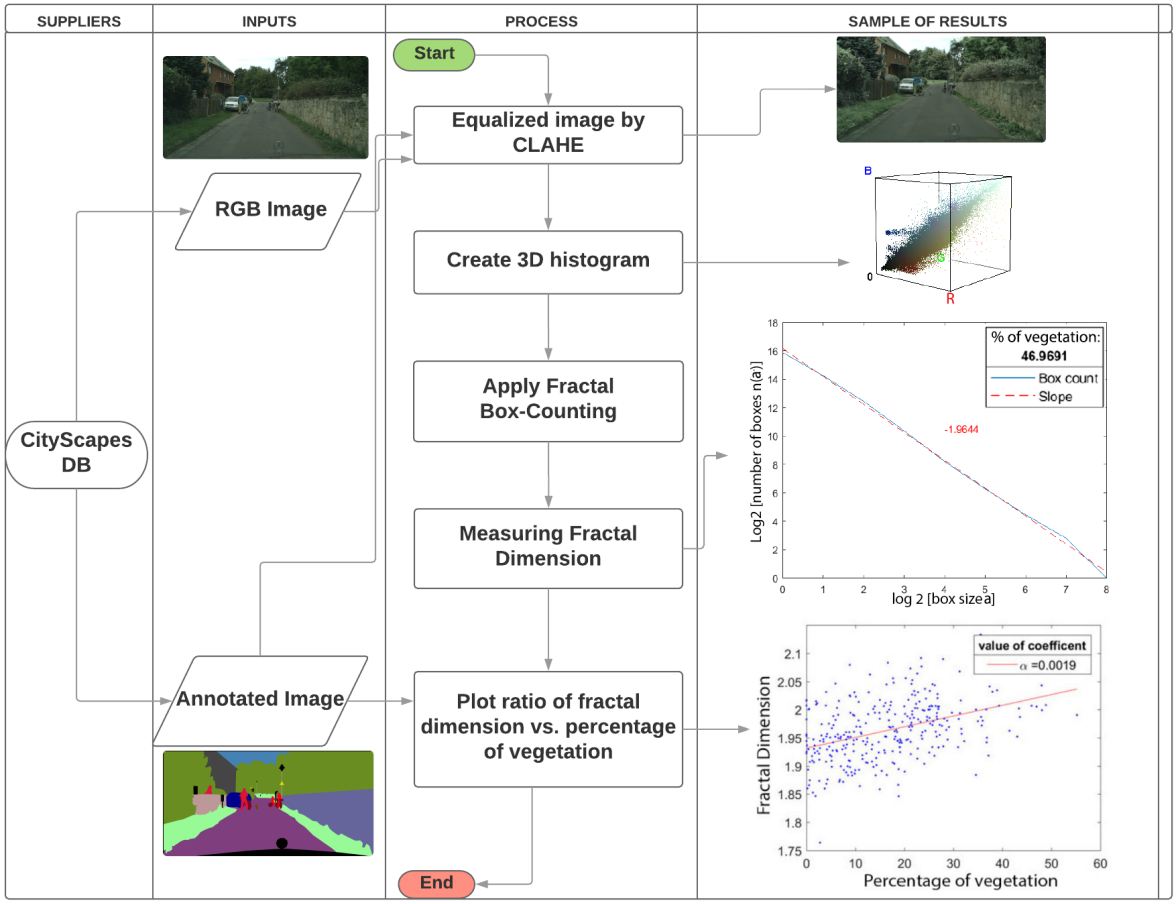


Figure 7. Image processing pipeline for the impact of vegetation in the color domain. CityScapes DB [30] corresponds to available dataset shows in Fig. 1. All images of CityScapes dataset are processed and batched city by city.

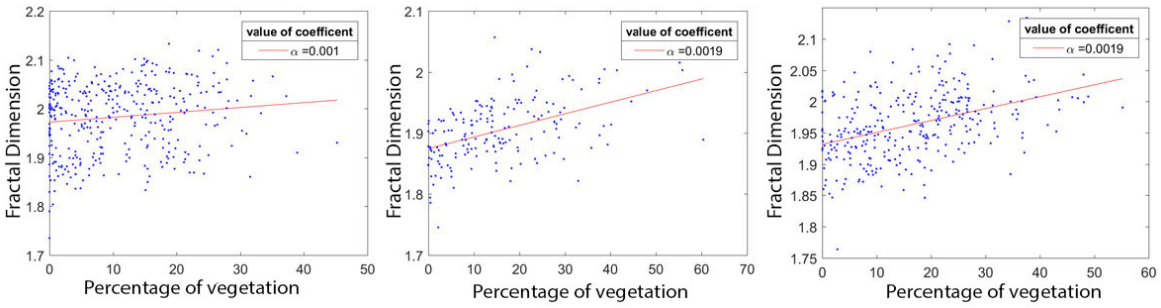


Figure 8. Fractal dimension values as a function of the percentage of the vegetation of three examples of cities taken from the dataset considered in this study [30]. The left column is Strasbourg city. The column in the middle is for Aachen. The right column is Bremen city.

Table 2. Slopes of fractal dimension and associated P-value in the colorimetric approach.

City	Slope	p-value
Aachen	0.19 E-2	3.47 E-11
Bochum	0.23 E-2	5.87 E-06
Bremen	0.19 E-2	1.59 E-14
Cologne	0.10 E-2	9.79 E-03
Darmstadt	0.35 E-2	1.45 E-09
Dusseldorf	0.23 E-2	5.67 E-14
Erfurt	0.17 E-2	3.94 E-07
Hamburg	0.18 E-2	1.54 E-06
Hanover	0.14 E-2	2.00 E-03
Jena	0.25 E-2	7.90 E-10
Krefeld	0.16 E-2	5.31 E-02
Monchengladbach	0.20 E-2	2.66 E-02
Strasbourg	0.10 E-2	1.35 E-02

Similarly to what was proposed in the previous section on Fourier, there are possible applications of this statistical result for urban planning. Especially the increase of the fractal dimension could serve to control a requested amount of vegetation in cities where the uniform colors of concrete are used for buildings. However, it is to be noticed that the complexity of the 3D color histogram can also be enriched in the direction of fractal signatures by simply painting concrete urban environment when there is no possibility of adding vegetation. This is shown in Fig. 9 on a colorful urban image without any vegetation. This image has a high fractal dimension which would correspond to adding almost 100% of vegetation in the mono-color concrete cities of Fig. 8.

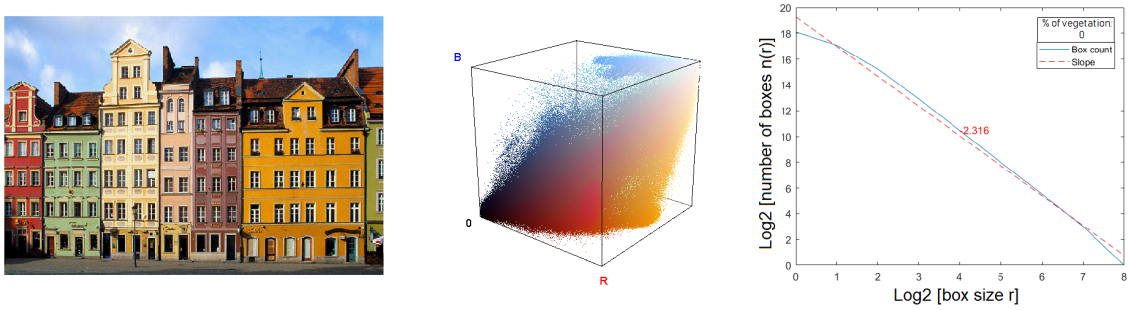


Figure 9. 3D color histogram and fractal dimension's plot of a colorful urban image without any vegetation.

4. Impact of vegetation on monocular depth cues

We explore the last aspect of how the impact of vegetation in pedestrian view can be quantified with computer vision tools and now focus on the depth, i.e. distance to a point of view. Humans and computers are known to be able to estimate depth from the stereo and monocular visions [38,39]. In this study, we restrict ourselves to monocular vision with single images of the streets. Different cues of depth can be present in monocular vision such as textures, shadows, defocus [40], parallel lines [41] producing in a projective geometry the presence of vanishing point on the horizon line, repetition of similar objects at different distances from the camera....

For humans, all these cues can contribute to the perception of depth in monocular vision and it is difficult to quantify the relative importance of each individual cue in a scene. With computers, it is possible to design a feature especially capturing the presence of a single depth cue. Also, with the current development of machine learning in computer vision, it is possible to produce a global quantitative estimation of depth incorporating all cues. In this section, we propose a quantification of the presence of vegetation on the quality of depth estimation with these two approaches including

Table 3. Available datasets for the analysis of the impact of vegetation on depth.

dataset	Num. of samples	RGB images	Annotated images	Depth images evaluated with
Make3D [48]	534	✓	-	Laser range data with ray position
KIITI [49]	93,000	✓	-	Lidar
Cityscapes [30]	25,000	✓	✓	SGM algorithm applied on RGB images [46]
WildDash [50]	70	✓	✓	-
Mapillary [51]	25,000	✓	✓	-
ApolloScape [47]	140,000	✓	✓	survey-grade dense 3D point cloud

the detection of the vanishing point and a direct estimation of the depth map from a monocular RGB image.

Virtual environments in urban systems research are very useful to access to situations that do not (yet) exist in real environments [42–44]. In our case, we need to have a dataset with RGB images of urban street view with various amounts of annotated vegetation. Ground truth depth map should not be computed from the RGB images to be compared with the depth cue estimation or depth estimation. There are a lot of available datasets with this structure for indoor and outdoor environments [45]. The most related outdoor datasets found in the literature are listed in table 3. It should be mentioned that for most of these datasets depths are not measured but estimated from RGB images in different ways such as semi-global matching (SGM) algorithm [46]. Therefore, they are not suitable depth for our purpose because we specifically want to extract depth from RGB and we need a depth map estimated from an independent distinct way as a ground truth. Also, the sole available outdoor dataset with a pedestrian urban street views in RGB and directly estimated depth [47] is not incorporating enough variation of percentage of vegetation. None of the available datasets listed in table 3 incorporate all the required aspects of our study. Therefore, we had to produce our own virtual dataset.

We propose the virtual RGBD green-city dataset provided as supplementary material to this study where 300 high-resolution images (879 $\times$ 1680 pixels) generated from the virtual world in urban settings under the different percentage of vegetation. These virtual cities were created using the Unity game engine with available models of trees and urban blocks. The dataset includes the segmentation of the vegetation to compute the percentage of vegetation and the depth map. Figure 10 illustrates the content of this virtual RGBD green-city dataset with three examples of RGB images with different percentages of vegetation. One specific interest in working with simulated data is that it is possible to create datasets of the same street with various amounts of plants following different strategies of the positioning of the plants. We take benefit of this opportunity and design several (10) experiments. This includes positioning trees on one side or on both sides of the street, positioning trees with various orientations or a single vertical orientation, using a tree with different sizes or using a variety of trees. For further information related to the virtual RGBD green-city dataset see supplementary material.

4.1. Detection of vanishing point

The vanishing point corresponds to the place where parallel edges in a real scene produce lines which cross in the image due to the projective geometry created by the lens of a camera. There are different approaches to find the vanishing points in an image [39,52,53], in this study our purpose is not to compare them or select any best method. Instead, we arbitrarily select one of the techniques which provides good results on similar images to ours and show how the performance of this technique evolves when the amount of vegetation is increased. To this purpose, the Hough

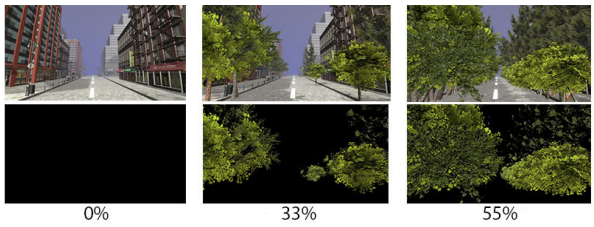


Figure 10. Three examples of virtual RGBD green-city dataset. The first row gives RGB images; the middle row is for the corresponding annotated images of the vegetation, the last row shows the percentage of vegetation.

transform-based technique of [54] is incorporated in the pipeline shown in Fig. 11.

A Hough transform is first applied to estimate the vanishing point on a reference image taken as the image with 0% of vegetation. Edges are extracted with the Canny edge detector [55] of the reference image. The extracted edges are then transformed into Hough space. The vanishing point is chosen as an intersection point with a large number (empirically chosen to 70) of intersection in the Hough space. Then, the extracted line segments are merged if they are associated with the same Hough transform bin and the distance between them is less than the value of a threshold empirically set to 400 pixels. Afterward, the detected lines in the Hough transform are counted as a contribution to the vanishing point if they are crossing in a vanishing area. Indeed, due to the limited and discretized size of the virtual environment, the lines do not intersect exactly at a single point [56]. We define the vanishing area with the size of 55 by 45 pixels around the largest number of convergent lines in the reference image without vegetation. Finally, the percentage of converged points in the vanishing area is recorded.

This pipeline (Fig.11) is applied to all the images of the dataset. The percentage of vanishing points is plotted with the comparison to the reference image free from vegetation as a function of the percentage of vegetation. Figure 12 shows the evolution of this percentage of vanishing points remaining as a function of the percentage of vegetation introduced. The values of the linear slope for all 10 experiments are represented in tables 4, 5. Table 4 represents slope values for scenes with trees located on both sides of the street or everywhere like a forest and table 5 shows slope values for scenes which trees are located just on one side of the street.

From Fig. 12 one can see that the presence of vegetation tends to change the position of vanishing points in the vanishing area and consequently fewer points are crossing in this area. An interpretation is that the increasing amount of leaf will increasingly cover the horizontal lines and thus affect the number of cues for the estimation of the vanishing point. However, it is to be noticed that other objects than vegetation could also occlude the horizontal lines and thus have an impact on the detection of vanishing points in a pure concrete urban environment. To further demonstrate the impact of vegetation on depth perception, we thus reproduced in the next section the experiment of this section using a completely different depth estimation method.

Table 4. Result for detected vanishing point in experiments with trees placed on both sides of the street and forest.

Category - two Side	Slope
Same Tree-Same Size-Same Orientation	- 0.9868
Same Tree-Same Size-Different Orientation	- 1.1056
Same Tree-Different Size-Same Orientation	- 0.5497
Different Tree-Same Size-Same Orientation	- 1.3513
Different Tree-Different Size-Different Orientation	- 0.9453
Forest	- 0.6916

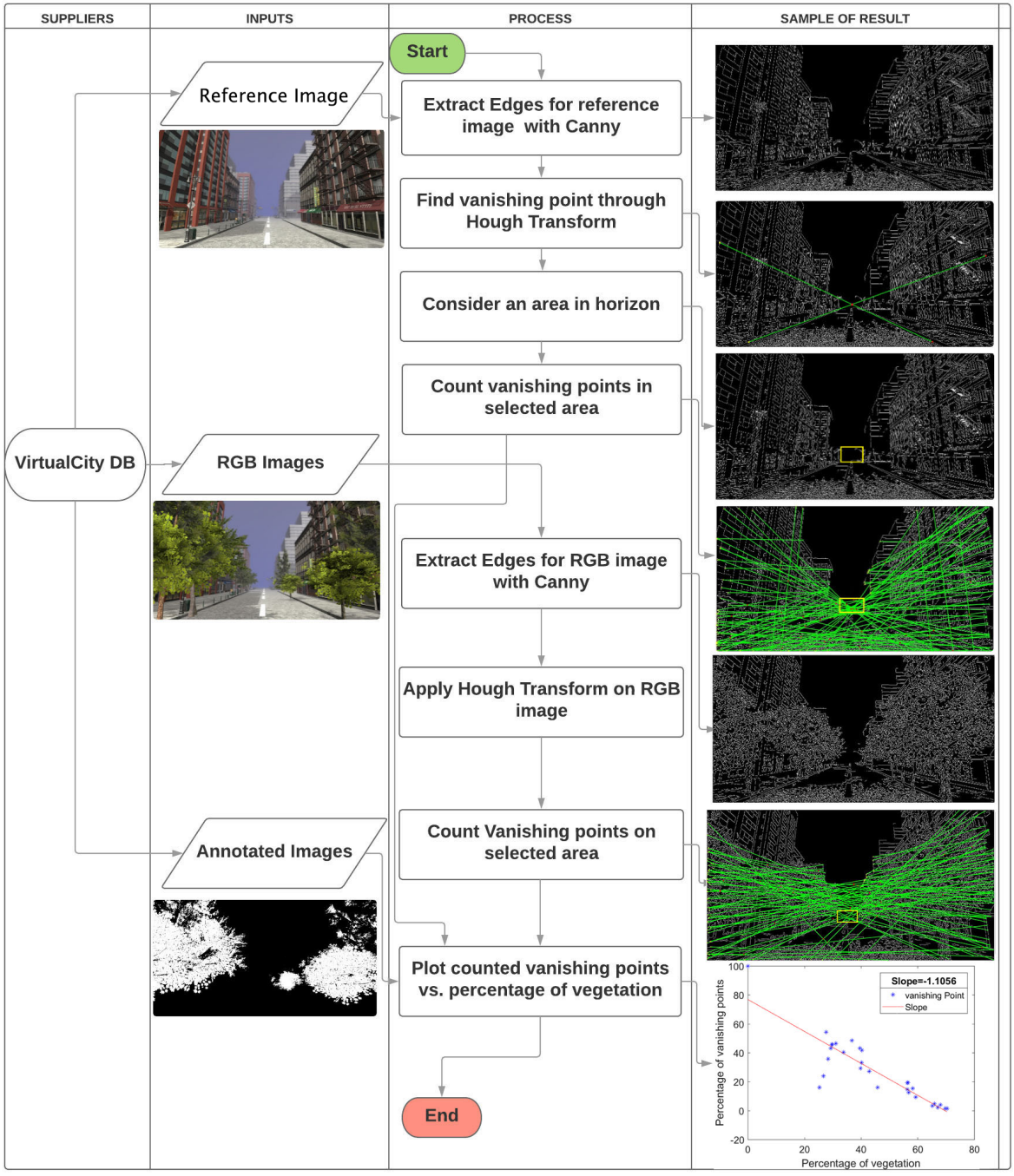


Figure 11. Image processing pipeline for the impact of vegetation on the detection of the vanishing point. VirtualCity DB refers to virtual RGBD green-city dataset which is provided for this study.

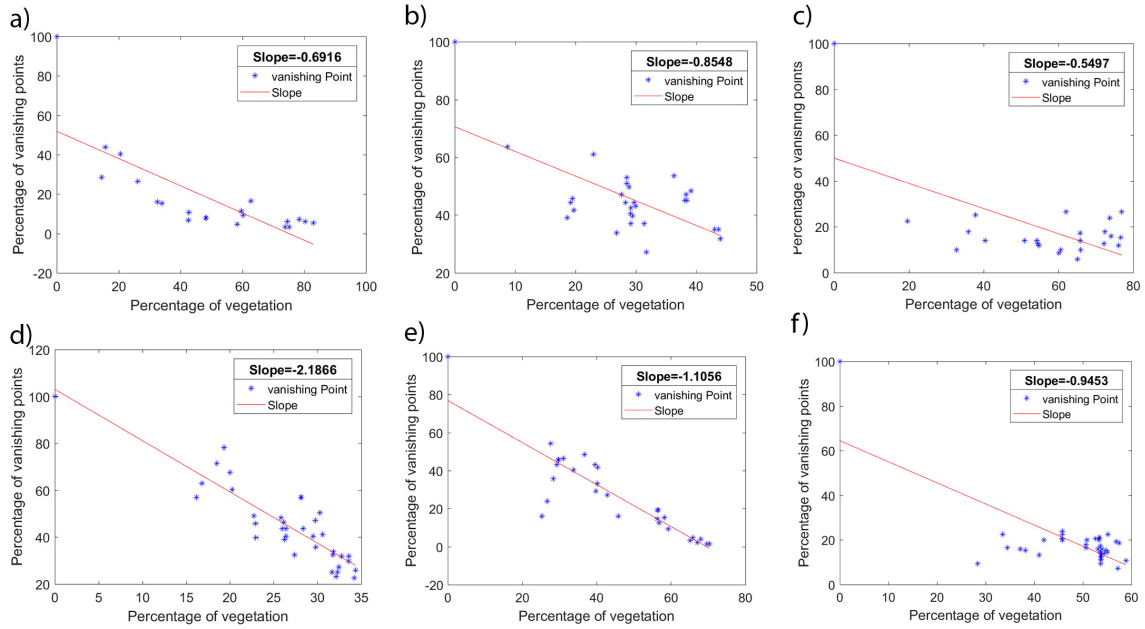


Figure 12. Percentage of detected vanishing points as a function of vegetation's percentage. a) Forest, b) Different-Size, One-Side c) Different-Size, Two-Sides d) Different-Orientation, One-Side e) Different-Orientation, Two-Side f) Different-Tree, Different-Size, Different-Orientation, Two-Side

Table 5. Result for detected vanishing point in experiments with trees placed on one side of the street.

Category - One Side	Slope
Same Tree-Same Size-Same Orientation	- 1.5522
Same Tree-Same Size-Different Orientation	- 2.1866
Same Tree-Different Size-Same Orientation	- 0.8548
Different Tree-Same Size-Same Orientation	- 1.3543

4.2. Depth estimation

We now propose to assess the impact of vegetation on the global perception of depth. To this purpose, we settled the pipeline described in Fig. 13. Here again, as for the estimation of the vanishing point, there is a huge literature on estimation of depth with machine learning approaches. Our purpose is not to compare any of them but to show the impact of vegetation on one of them [57]. We arbitrarily select one of the recent methods performing well for depth estimation of similar images to ours and submit this to our original virtual RGBD green-city dataset which contains RGB, depth and annotated images.

At the first step, the estimated depth is calculated by Deep Convolutional Neural Field (DCNF) algorithm developed by [57]. DCNF model is a depth estimation approach exploring Convolutional Neural Network (CNN) and continuous Conditional Random Field (CRF). The whole networks of DCNF are trained on Make3D dataset [48] for outdoor scenes. Make3D dataset approximately includes 1000 outdoor street pedestrian views captured in good weather conditions with 50% of vegetation on average. At the second step, the global similarity is calculated between the estimated depth images and ground-truth depth by normalized-cross-covariance (NCC) which reads

$$NCC(X,Y) = \frac{\mathbf{E}\left[((X-\mathbf{E}(X)).(Y-\mathbf{E}(Y))\right]}{\sigma_X \sigma_Y}, \tag{2}$$

where X and Y represent the estimated depth map and ground-truth depth map respectively, σ_X and σ_Y are the standard deviations and **E** stands for the 2D mean.

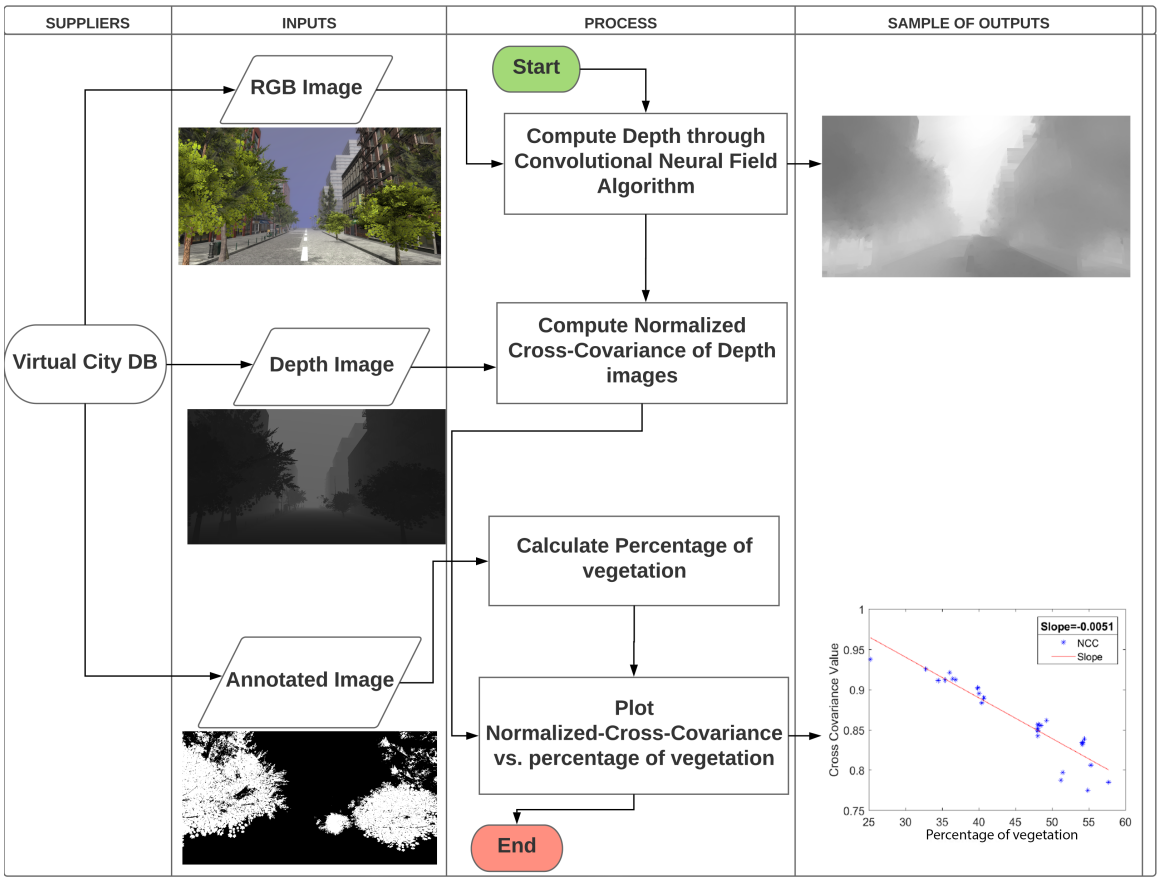


Figure 13. Image processing pipeline for the impact of vegetation on normalized-cross-covariance approach. VirtualCity DB refers to virtual RGBD green-city dataset which is provided for this study.

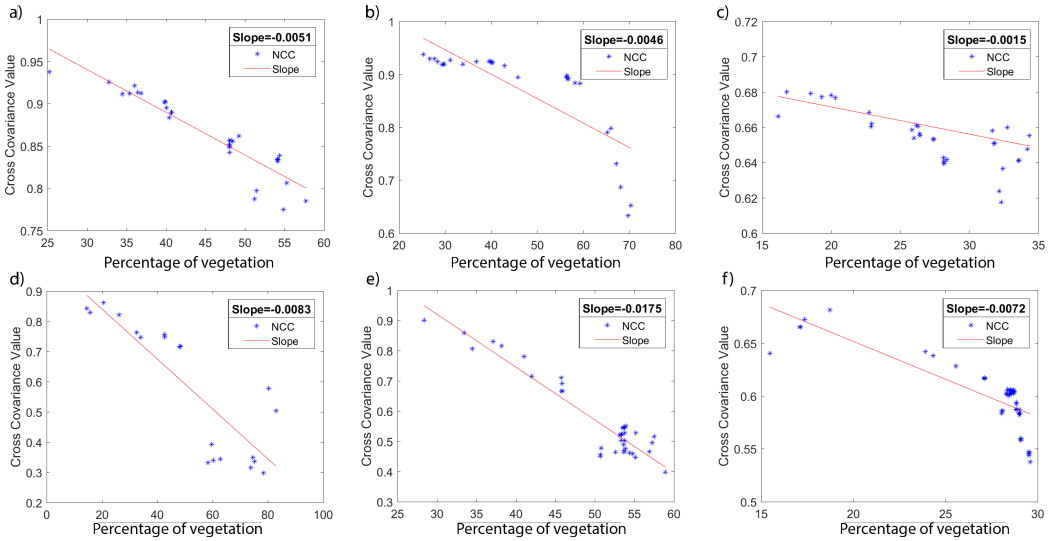


Figure 14. Normalized cross-covariance as a function of the percentage of vegetation. a) Same tree-same size-Same orientation-Two side, b) Different orientation-Two side, c) Different orientation-One side, d) Forest, e) Different tree-Different size-Different orientation-Two side, f) Same tree-Same size-Same orientation-One side

Figure 14 illustrates the value of normalized-cross-covariance as a function of the percentage of vegetation for the six different experiments included in the virtual RGBD green-city dataset. The impact of vegetation on the similarity of estimated depth map with the true depth map is systematic. The presence of the vegetation tends to decrease the quality of depth cues. This decrease modeled with a linear trend gives the slope provided in table 6 for scenes with trees located on both sides of the street or everywhere like a forest and table 7 for scenes with trees located just on one side of the street. Interestingly this result is similar to the one observed with the Hough transform in the previous section while it was obtained with a completely different approach.

From an applied urban planning perspective, if we discard the solution that would correspond to position trees everywhere (Forest case) these experiments demonstrate that the highest decreases of depth cue (as given in table 4-7) are obtained for urban streets when they include a variability of tree shape and/or tree size and/or various tree orientations. Projective geometry with parallel lines like the one produced by urban block world is not present in wild landscapes. Therefore, we come up with the interesting conclusion that, consistently with the well-known necessity of diversity in ecosystems, computer vision also suggests to add plant diversity as the most effective strategy to break the depth cues created by non-natural urban block worlds.

Table 6. Result for global similarity in experiments with trees placed on both sides of the street and forest.

Category - two Side	Slope
Same Tree-Same Size-Same Orientation	- 0.0072
Same Tree-Same Size-Different Orientation	- 0.0015
Same Tree-Different Size-Same Orientation	- 0.0089
Different Tree-Same Size-Same Orientation	- 0.0019
Different Tree-Different Size-Different Orientation	- 0.0175
Forest	- 0.0083

Table 7. Result for global similarity in experiments with trees placed one side of the street.

Category - One Side	Slope
Same Tree-Same Size-Same Orientation	- 0.0051
Same Tree-Same Size-Different Orientation	- 0.0046
Same Tree-Different Size-Same Orientation	- 0.0118
Different Tree-Same Size-Same Orientation	- 0.0016

5. Conclusion

In this pilot experimental study, we quantified, on large datasets, the impact of vegetation on visually perceptible symmetries in urban street pedestrian viewpoint. Correlation of this amount of vegetation with objective computer vision traits has been shown statistically in the Fourier domain, in the color histogram, and in depth from monocular view. This objectively quantifies the expected common-sense intuition that vegetation in street pedestrian views breaks the orthogonal symmetry of urban blocks, enriches the color space in the direction of higher dimension fractal symmetries and decreases the cues of depth included in projective geometry. The result obtained in the Fourier domain and the color histogram corresponded to existing experiments of the literature carried here for the first time on urban scene with various amount of vegetation while the experiment designed and carried on depth is, to the best of our knowledge, completely new.

Possible applications in urban planning of the carried experiment have been proposed. The most interesting is that a percentage of the vegetation of 20 to 30% is found to be necessary to have an urban street which appear closer to natural images than pure man-made from the Fourier point of view. Interestingly this also meets the typical percentage of vegetation often mentioned as necessary to maintain ecological viability and diversity in urban ecosystemic studies [32–34]. The novel contribution on the impact of vegetation on depth could be extended in various directions to enable a deeper understanding of the recorded effect. For instance, it would be possible to reproduce the experiment carried here to objectively quantify the effect of vegetation on other computer vision traits. In particular, one could investigate the pedestrian viewpoint in urban streets with bio-inspired traits such as the estimation of depth from stereovision [58–60] or at a higher integrative level from visual saliency, [61] or also with the heat map produced by eye-tracking systems when applied on large cohorts of human observers [62]. The effect of vegetation on pedestrian view in urban street was objectively quantified in this article. It could also be interesting to compare these results with aesthetic assessments [63–65] on the same scenes when perceived by the human. One could directly embed humans in virtual environments with grabbing tasks (similarly to [66] for instance) and assess their performance while varying the amount of vegetation. Finally, with the analytical approach followed in this preliminary work, the standard deviation from the simple linear trends identified on few descriptors, although already interesting, could be considered as still too high to serve as good predictors of a correct amount of vegetation to be placed in an urban planning. Using more expressive multivariate models with a learned and large feature space based on deep learning [67] could be a promising direction.

In another direction, one could seek to apply the computer vision traits used in this study on cognitive architecture [68] or urban landscape experiments carried in the domain of ecology. In architecture, one could, for instance, investigate pedestrian view of the urban environment through the window at different floors of buildings (while the experiment in this article was limited to ground floor view). In the domain of ecology, one could study the relationship between the computer vision metrics designed in this study and the ecological diversity along the seasons (while the experiment in this article was limited to spring-like views with unlabeled ecological diversity) or in night vision as allowed by the simulation environment introduced in this work. Such architectural or ecological experiments had in the past been relying on qualitative psycho-visual studies [69,70] but also tend to use computer vision features in 2D dimensions [71,72] or as recently shown in [73] with 3D LIDAR data.

6. Supplementary material

As a side result of this work, a novel dataset has been produced which corresponds to a synthetic urban street in pedestrian view through RGB images with the various amount of vegetation. These images are associated with depth maps and percentages of vegetation. It is made freely available as pointed in the supplementary material section.

Author Contributions: Conceptualization, S.S. and D.R.; methodology, S.S., P.R. and D.R; software, S.S. and P. R; validation, P.R., D.R. and H. D.; resources, P.R.; data curation, S.S.; writing—original draft preparation, S.S. and D. R.; writing—review and editing, S.S., D.R, E. B. and H. D.; visualization, S.S.; supervision, D.R.; project administration, D.R.; funding acquisition, D.R. and P. R.

Funding: Salma Samiei acknowledges Angers Loire Métropole for the funding of her PhD.

1. Wolch, J.R.; Byrne, J.; Newell, J.P. Urban green space, public health, and environmental justice: The challenge of making cities 'just green enough'. *Landscape and Urban Planning* **2014**, *125*, 234–244.
2. Li, X.; Zhang, C.; Li, W.; Ricard, R.; Meng, Q.; Zhang, W. Assessing street-level urban greenery using Google Street View and a modified green view index. *Urban Forestry & Urban Greening* **2015**, *14*, 675–685.
3. Carpenter, M. From 'healthful exercise'to 'nature on prescription': The politics of urban green spaces and walking for health. *Landscape and Urban Planning* **2013**, *118*, 120–127.
4. Coppel, G.; Wüstemann, H. The impact of urban green space on health in Berlin, Germany: Empirical findings and implications for urban planning. *Landscape and Urban Planning* **2017**, *167*, 410–418.
5. Ekkel, E.D.; de Vries, S. Nearby green space and human health: Evaluating accessibility metrics. *Landscape and Urban Planning* **2017**, *157*, 214–220.
6. Sugiyama, T.; Carver, A.; Koohsari, M.J.; Veitch, J. Advantages of public green spaces in enhancing population health. *Landscape and Urban Planning* **2018**, *178*, 12–17.
7. Russo, A.; Cirella, G. Modern compact cities: how much greenery do we need? *International journal of environmental research and public health* **2018**, *15*, 2180.
8. du Toit, M.J.; Cilliers, S.S.; Dallimer, M.; Goddard, M.; Guenat, S.; Cornelius, S.F. Urban green infrastructure and ecosystem services in sub-Saharan Africa. *Landscape and Urban Planning* **2018**.
9. Zinia, N.J.; McShane, P. Ecosystem services management: An evaluation of green adaptations for urban development in Dhaka, Bangladesh. *Landscape and Urban Planning* **2018**, *173*, 23–32.
10. Chen, B.; Adimo, O.A.; Bao, Z. Assessment of aesthetic quality and multiple functions of urban green space from the users' perspective: The case of Hangzhou Flower Garden, China. *Landscape and Urban Planning* **2009**, *93*, 76–82.
11. Salesses, P.; Schechtner, K.; Hidalgo, C.A. The collaborative image of the city: mapping the inequality of urban perception. *PloS one* **2013**, *8*, e68400.
12. Liu, L.; Silva, E.A.; Wu, C.; Wang, H. A machine learning-based method for the large-scale evaluation of the qualities of the urban environment. *Computers, Environment and Urban Systems* **2017**, *65*, 113–125.
13. Li, X.; Ratti, C.; Seiferling, I. Quantifying the shade provision of street trees in urban landscape: A case study in Boston, USA, using Google Street View. *Landscape and Urban Planning* **2018**, *169*, 81–91.
14. Li, X.; Zhang, C.; Li, W.; Kuzovkina, Y.A. Environmental inequities in terms of different types of urban greenery in Hartford, Connecticut. *Urban Forestry & Urban Greening* **2016**, *18*, 163–172.
15. Long, Y.; Liu, L. How green are the streets? An analysis for central areas of Chinese cities using Tencent Street View. *PloS one* **2017**, *12*, e0171110.
16. Small, C. Estimation of urban vegetation abundance by spectral mixture analysis. *International journal of remote sensing* **2001**, *22*, 1305–1334.
17. Antczak, E. Urban Greenery in the Greatest Polish Cities: Analysis of Spatial Concentration. *World Academy of Science, Engineering and Technology, International Journal of Transport and Vehicle Engineering* **2017**, *4*.
18. McCool, C.; Beattie, J.; Milford, M.; Bakker, J.D.; Moore, J.L.; Firn, J. Automating analysis of vegetation with computer vision: Cover estimates and classification. *Ecology and Evolution* **2018**.
19. Zhaoping, L.; Li, Z. *Understanding vision: Theory, models, and data*; Oxford University Press, USA, 2014.

20. Elder, J.H.; Victor, J.; Zucker, S.W. Understanding the statistics of the natural environment and their implications for vision. *Vision research* **2016**, *120*, 1–4.
21. De Cesarei, A.; Loftus, G.R.; Mastria, S.; Codispoti, M. Understanding natural scenes: Contributions of image statistics. *Neuroscience & Biobehavioral Reviews* **2017**.
22. Chéné, Y.; Belin, É.; Rousseau, D.; Chapeau-Blondeau, F. Multiscale analysis of depth images from natural scenes: Scaling in the depth of the woods. *Chaos, Solitons & Fractals* **2013**, *54*, 135–149.
23. Adams, W.J.; Elder, J.H.; Graf, E.W.; Leyland, J.; Lugtigheid, A.J.; Muryy, A. The southampton-york natural scenes (syns) dataset: Statistics of surface attitude. *Scientific reports* **2016**, *6*, 35805.
24. Morris, N.J.; Avidan, S.; Matusik, W.; Pfister, H. Statistics of infrared images. Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007, pp. 1–7.
25. Vaughn, I.J.; Alenin, A.S.; Tyo, J.S. Statistical scene generation for polarimetric imaging systems. *arXiv preprint arXiv:1707.02723* **2017**.
26. Balboa, R.M.; Grzywacz, N.M. Power spectra and distribution of contrasts of natural images from different habitats. *Vision research* **2003**, *43*, 2527–2537.
27. Rosch, E.; Mervis, C.B.; Gray, W.D.; Johnson, D.M.; Boyes-Braem, P. Basic objects in natural categories. *Cognitive psychology* **1976**, *8*, 382–439.
28. Torralba, A.; Oliva, A. Statistics of natural image categories. *Network: computation in neural systems* **2003**, *14*, 391–412.
29. Samavatekbatan, A.; Gholami, S.; Karimimoshaver, M. Assessing the visual impact of physical features of tall buildings: Height, top, color. *Environmental Impact Assessment Review* **2016**, *57*, 53–62.
30. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
31. Reza, A.M. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. *The Journal of VLSI Signal Processing* **2004**, *38*, 35–44.
32. Lindenmayer, D.; Luck, G. Synthesis: thresholds in conservation and management. *Biological Conservation* **2005**, *124*, 351–354.
33. Fahrig, L. Effect of habitat fragmentation on the extinction threshold: a synthesis. *Ecological applications* **2002**, *12*, 346–353.
34. Huggett, A.J. The concept and utility of 'ecological thresholds' in biodiversity conservation. *Biological conservation* **2005**, *124*, 301–310.
35. Chapeau-Blondeau, F.; Chauveau, J.; Rousseau, D.; Richard, P. Fractal structure in the color distribution of natural images. *Chaos, Solitons & Fractals* **2009**, *42*, 472–482.
36. Chauveau, J.; Rousseau, D.; Chapeau-Blondeau, F. Fractal capacity dimension of three-dimensional histogram from color images. *Multidimensional Systems and Signal Processing* **2010**, *21*, 197–211.
37. Chauveau, J.; Rousseau, D.; Richard, P.; Chapeau-Blondeau, F. Multifractal analysis of three-dimensional histogram from color images. *Chaos, Solitons & Fractals* **2010**, *43*, 57–67.
38. Torralba, A.; Oliva, A. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence* **2002**, *24*, 1226–1238.
39. Szeliski, R. *Computer vision: algorithms and applications*; Springer Science & Business Media, 2010.
40. Ziou, D.; Deschenes, F. Depth from defocus estimation in spatial domain. *Computer vision and image understanding* **2001**, *81*, 143–165.
41. Rogez, G.; Orrite, C.; Guerrero, J.; Torr, P.H. Exploiting projective geometry for view-invariant monocular human motion analysis in man-made environments. *Computer Vision and Image Understanding* **2014**, *120*, 126–140.
42. Portman, M.E.; Natapov, A.; Fisher-Gewirtzman, D. To go where no man has gone before: Virtual reality in architecture, landscape architecture and environmental planning. *Computers, Environment and Urban Systems* **2015**, *54*, 376–384.
43. Kuliga, S.F.; Thrash, T.; Dalton, R.C.; Hoelscher, C. Virtual reality as an empirical research tool—Exploring user experience in a real building and a corresponding virtual model. *Computers, Environment and Urban Systems* **2015**, *54*, 363–375.
44. Fisher-Gewirtzman, D.; Portman, M.; Natapov, A.; Hölscher, C. Special electronic issue:"The use of virtual reality for environmental representations". *Computers, environment and urban systems* **2017**, *62*, 97–98.

45. Stamos, I.; Pollefeys, M.; Quan, L.; Mordohai, P.; Furukawa, Y. Special Issue on Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans. *Computer Vision and Image Understanding* **2017**, pp. 1–2.
46. Hirschmuller, H. Accurate and efficient stereo processing by semi-global matching and mutual information. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, Vol. 2, pp. 807–814.
47. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The ApolloScape Dataset for Autonomous Driving. *arXiv preprint arXiv:1803.06184* **2018**.
48. Saxena, A.; Chung, S.H.; Ng, A.Y. Learning depth from single monocular images. Advances in neural information processing systems, 2006, pp. 1161–1168.
49. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* **2013**.
50. Zendel, O.; Murschitz, M.; Humenberger, M.; Herzner, W. How Good Is My Test Data? Introducing Safety Analysis for Computer Vision. *International Journal of Computer Vision* **2017**, *125*, 95–109.
51. Neuhold, G.; Ollmann, T.; Bulò, S.R.; Kontschieder, P. The mapillary vistas dataset for semantic understanding of street scenes. Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 22–29.
52. Simond, N.; Rives, P. Homography from a vanishing point in urban scenes. Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. IEEE, 2003, Vol. 1, pp. 1005–1010.
53. Zhou, Z.; He, S.; Li, J.; Wang, J.Z. Modeling perspective effects in photographic composition. Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 301–310.
54. Li, B.; Peng, K.; Ying, X.; Zha, H. Vanishing point detection using cascaded 1D Hough Transform from single images. *Pattern Recognition Letters* **2012**, *33*, 1–8.
55. Canny, J. A computational approach to edge detection. In *Readings in Computer Vision*; Elsevier, 1987; pp. 184–203.
56. Chang, H.; Tsai, F. Reconstructing Three-Dimensional Specific Curve Building Models from a Single Perspective View Image. *Ternational Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2012**, *39*, 101–106.
57. Liu, F.; Shen, C.; Lin, G.; Reid, I. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence* **2016**, *38*, 2024–2039.
58. Bertamini, M.; Martinovic, J.; Wuerger, S.M. Integration of ordinal and metric cues in depth processing. *Journal of Vision* **2008**, *8*, 10–10.
59. Rzeszutek, R.; Androutsos, D. A framework for estimating relative depth in video. *Computer Vision and Image Understanding* **2015**, *133*, 15–29.
60. Turski, J. The conformal camera in modeling active binocular vision. *Symmetry* **2016**, *8*, 88.
61. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* **1998**, *20*, 1254–1259.
62. Duchowski, A.T. Eye tracking methodology. *Theory and practice* **2007**, *328*.
63. Rigau, J.; Feixas, M.; Sbert, M. Informational aesthetics measures. *IEEE Computer Graphics and Applications* **2008**, *28*.
64. Dresp-Langley, B. Affine geometry, visual sensation, and preference for symmetry of things in a thing. *Symmetry* **2016**, *8*, 127.
65. Chen, C.C.; Wu, J.H.; Wu, C.C. Reduction of image complexity explains aesthetic preference for symmetry. *Symmetry* **2011**, *3*, 443–456.
66. Batmaz, A.U.; de Mathelin, M.; Dresp-Langley, B. Effects of Image Size and Structural Complexity on Time and Precision of Hand Movements in Head Mounted Virtual Reality. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2018, pp. 167–174.
67. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep learning*; Vol. 1, MIT press Cambridge, 2016.
68. Sussman, A.; Hollander, J.B. *Cognitive architecture: designing for how we respond to the built environment*; Routledge, 2014.
69. Zacharias, J. Preferences for view corridors through the urban environment. *Landscape and urban planning* **1999**, *43*, 217–225.

70. Danahy, J.W. Technology for dynamic viewing and peripheral vision in landscape visualization. *Landscape and Urban Planning* **2001**, *54*, 127–138.
71. Stamps, A.E. Fractals, skylines, nature and beauty. *Landscape and urban planning* **2002**, *60*, 163–184.
72. Van den Berg, A.E.; Joye, Y.; Koole, S.L. Why viewing nature is more fascinating and restorative than viewing buildings: A closer look at perceived complexity. *Urban forestry & urban greening* **2016**, *20*, 397–401.
73. Casalegno, S.; Anderson, K.; Hancock, S.; Gaston, K.J. Improving models of urban greenspace: from vegetation surface cover to volumetric survey using waveform laser scanning. *Methods in Ecology and Evolution* **2017**.

Chapter 6

Annex B

Machine Learning-Based Classification of the Health State of Mice Colon in Cancer Study from Confocal Laser Endomicroscopy

Pejman Rasti[1], Christian Wolf[2], Hugo Dorez[3], Raphael Sablong[3], Driffa Moussata[3], Salma Samiei[1], and David Rousseau[1,*]

[1]Laboratoire Angevin de Recherche en Ingénierie des Système (LARIS),UMR INRA IRHS, Université d'Angers, Angers, France.
[2]INSA-Lyon, INRIA, LIRIS, CITI, CNRS, Villeurbanne, France
[3]CREATIS, Université Lyon 1, Villeurbanne, France
[*]david.rousseau@univ-angers.fr

ABSTRACT

In this article, we address the problem of the classification of the health state of the colon's wall of mice, possibly injured by cancer with machine learning approaches. This problem is essential for translational research on cancer and is a priori challenging since the amount of data is usually limited in all preclinical studies for practical and ethical reasons. Three states considered including cancer, health, and inflammatory on tissues. Fully automated machine learning-based methods are proposed, including deep learning, transfer learning, and shallow learning with SVM. These methods addressed different training strategies corresponding to clinical questions such as the automatic clinical state prediction on unseen data using a pre-trained model, or in an alternative setting, real-time estimation of the clinical state of individual tissue samples during the examination. Experimental results show the best performance of 99.93% correct recognition rate obtained for the second strategy as well as the performance of 98.49% which were achieved for the more difficult first case.

Introduction

Classically the characterization of colon's pathology is realized from histology[1] but is now also investigated with in vivo imaging techniques which enable the oncological[2] early detection of abnormal physiological processes such as inflammation of dysplastic lesions. This includes chromoendoscopy[3], confocal laser endomicroscopy[4,5] or multiphoton microscopy[6]. These modern video-microscopies introduced in preclinical studies on mice with the promises of translational research[7].
These imaging techniques are producing videos which for the inspection of one colon of one mouse corresponds to thousands of frames to be further multiplied by the number of mice inspected. Each frame of these videos can be different in the structure and texture as it is recorded over a colon's wall with movement of the probe, spurious presence of unexpected items between probes and colon, variation of contrast agent concentration. To draw benefit from such imaging protocols, the bottleneck is thus the automation of the image analysis. In this article, we consider one of these protocols and propose a fully automated solution for the classification of colon wall images into healthy, inflammation and dysplastic tissues.
We work with the confocal endomicroscopy imaging protocol of[5] for the classification of the health state of the colon's wall of mice. Since its introduction, this protocol has seen widespread usage in multiple research groups[8–10]. So far, image analysis for the classification of colon's wall health state with this protocol has been relatively limited. The existing literature is based on handcrafted features[5,8–10].

In this article, we go beyond the sole characterization (feature handcrafting) and, for the first time on Mice colon in cancer study from confocal laser endomicroscopy, in the growing trend of machine learning applied to medical image analysis[11–13], propose a fully automated classification method based on supervised learning that we validate on thousands of images. This work is a priori challenging since the amount of data in preclinical studies, such as in our case, is rather limited compared to the usual amount of data available in medical applications of machine learning. Also, another a priori open question addressed in the preclinical study is the question of translational research, i. e. the reusability of the knowledge gained for animals on human or human on animals. We address this question here, for the first time to our knowledge, in the perspective of machine learning. As the last innovation in our methodology to address a specific unsolved preclinical problem, we discuss different scientific use cases and corresponding strategies for training concerning some properties of confocal laser endomicroscopy.

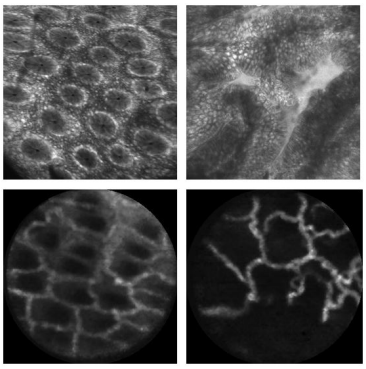


Figure 1. Top: Human samples of colon's wall images: healthy (left) and unhealthy (right) tissues observed from fluorescent confocal endomicroscopy. Bottom: Mouse samples of colon's wall images: healthy (left) and unhealthy (right) tissues observed from fluorescent confocal endomicroscopy.

Images are acquired at the video frame rate while the expert holding the endoscopic probes moves it slowly to inspect the tissue when located close to the tissue of interest. Consequently, though the imaging system is producing vast amounts of images, a large number of images are very similar. We consider the possibility of taking benefit from this self-similarity in order to significantly reduce the size of the data set requested during the training stage. This training approach is vital for the expert in charge of the annotation of the training data sets since it is a highly time-consuming task. In a second configuration, we also discuss the performance obtained with different machine learning approaches when we learn on images corresponding to a given set of mice while applying the classification on a distinct cohort of mice. This cross-subject training is relevant for clinical purposes because it quantifies to which extend the disease observed is generic or patient-specific. The performances of these two training strategies compared to the best performance obtained with a brute force random sampling on a whole cohort for the training of the classification algorithm.

In the literature, several studies have focused on the classification of colon's health state from endomicroscopy. Up to our knowledge, this body of work based on the classical methodology of handcrafted feature design (taking into account domain knowledge), followed by supervised machine learning.
A method based on global descriptors proposed in[5], whose introduced fractal box-counting metrics and illustrated them on two images. Vessel detection was proposed in[8] after a Hessian-based filter in addition to length area and diameter measurements of vascular crypts of the colon's wall. Blood vessels of the colon's wall characterized in[9] from Fourier analysis. Also, vascular networks of colon's wall were characterized in terms of graphs in[10] after skeletonization on few hundreds of images.
Closest to our work is the method by Ştefănescu et al., which is based on machine learning with neural networks of images of human tissues[14] acquired with confocal laser endomicroscopy. However, the images are clearly different; in contrast, the field of view and resolution, as can be seen in Fig. 1. These differences motivate our proposition of designing a specific method for mice trained on mouse images. In contrast to[14], we (i) propose a method based on representation learning[15] as opposed to handcrafted features, and (ii) specifically discuss different experimental protocols and develop different training strategies adapted to these protocols.

Results

In this section, we give experimental results using the experimental protocol and training strategies described in the method section as well as the different feature extraction and feature learning techniques.

Cross-subject training

For this protocol, the most challenging one of all considered cases, where generalization to unseen subjects (mice) is required, randomly chosen images of mice for three datasets of training, validation, and testing as shown in table 1. While the training set is used to adjust the parameters of the model, the validation set is used to minimize overfitting and tune the parameters. The test set of unseen data is used to confirm the predictive power and that the model generalises. The final classification of trials is computed as the average performance of each fold. The number of healthy and unhealthy mice are not equal. We simulated cross-validation for this approach by changing mice between training, validation, and testing for each new experiment.
Table 2 gives results with the different feature representations and classifiers described in the method section. In addition, table 3 shows classification accuracy of a transfer learning method with different freezing layers discussed in section . Our

Table 1. Number of mice in each dataset

	Healthy mice	Mice with cancer	Mice with inflammation
Training	5	7	7
Validation	1	2	2
Testing	3	4	7

proposed architecture trained from scratch shows the best recognition rate compared to handcrafted features, and state of the art high-capacity architectures with pre-training. The experiments indicate that high-capacity networks overfit on this amount of target data even when they are pre-trained on large datasets of natural images. We conjecture that the shift in data distributions is too large in the case of this application. The last layer of the network, still trained from scratch even in the case of transfer learning, overfits on the small target data set. To sum up the essence of the contribution, we train a high-capacity model on a large scale data set, followed by fine-tuning of a low capacity SVM model on the small volume target data set.
Also, we studied the dependency of the classification results on the number of subjects in the training data, as illustrated in the figure 2. For this study, we chose the LBP based representation and the SVM classifier since it can work better when a small size of the database is available for training. As expected, the system performance increases significantly when additional mice are added to the training set, as each mouse potentially has its specific pattern for health, inflammation, and cancer tissues.
Figure 3 shows some cases of correctly and wrongly classified images with their coarse localization maps. As can be seen, these images are indeed difficult to assess as the miss classified images have a similar pattern with another class.

Table 2. Left: Results of cross-subject training with full data, where all images of 6 healthy mice, 9 mice with cancer, and 9 mice with inflammation used for training the system. Right: Confusion matrix of cross-subject performance where our proposed CNN architecture is used.

Classifiers	Transfer learning	Accuracy
Proposed CNN architecture	-	**98.49**% ±0.6
DenseNet	X	94.54% ±2.9
VGG16 + linear SVM	X	90.60% ±0.4
VGG16	X	89.62 % ±3.3
ResNet50	X	75.93% ±4.1
VGG16	-	74.82% ±3.2
LBP features + linear SVM	-	83.01% ±0.4
Proposed method at[14]	-	77.41% ±1.3

	True Cancer	True Inflammation	True Healthy
Predicted Cancer	13107	0	0
Predicted Inflammation	0	5012	46
Predicted Healthy	0	75	2011

Table 3. Results of cross-subject training with different numbers of frozen layers when transferring the VGG16 network from ImageNet to the target dataset.

No. Freezing Conv. layers	1	2	3	4	5	6	7	8	9	10	11	12	13
Accuracy	40.8%±17.4	65.6±29.9%	89.6±3.3%	89.2%±3.9	42.8%±21.9	43.4%±23.25	70%±24.1	52.8%±22.2	75.4%±23.9	82.2%±9.4	65.8%±29.9	41.2%±18.3	33%±0

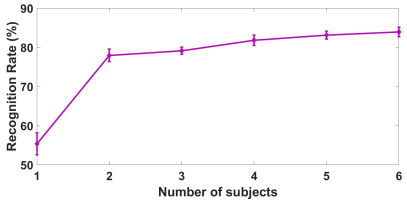


Figure 2. Dependency on the number of training subjects for cross-subject training (LBP features + SVM classifier).

Cross-sample training with all samples

Let us recall that in another use case of cross-sample training, subjects (mice) are mixed between training and test sets. In our setup, the 7 fold cross-validation approach used where almost 75% of images are dedicated for training and 25% of images for testing purposes, which corresponds to the proportions chosen for a similar problem in[14], albeit for human colon's walls. When

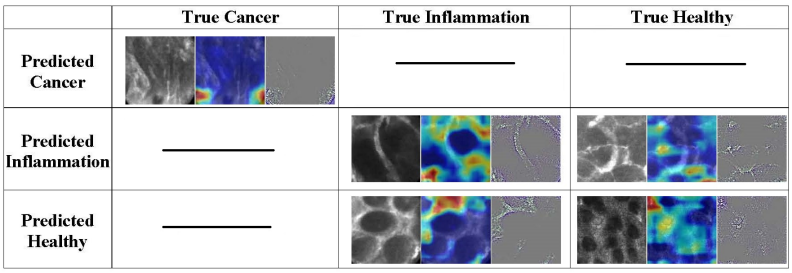


Figure 3. Example of correctly and miss classified images of the proposed CNN architecture for the cross-subject training strategy. Each cell consists from left to right of a grayscale image, a coarse localization map of the important regions in the image for the network[16], and a high-resolution class-discriminative visualization[16]. Cells with dashed lines mean that there is no miss classified images for that class.

needed, the validation set was chosen from the training set. Table 4 gives the prediction performance of the different classifiers on this data. We report means and standard deviations of ten runs.

Table 4. Left: Results of cross-sample training with full data. Right: Confusion Matrix of cross-sample performance where our proposed CNN architecture is used.

Classifiers	Transfer learning	Accuracy
Proposed CNN architecture	-	**99.93%** ±0.13
LBP features + linear SVM	-	97.7% ±0.39
VGG16 + linear SVM	X	85.9% ±0.4
VGG16	X	82.12% ±4.1
ResNet50	X	79.94% ±4.6
DenseNet	X	79.51% ±3.8
VGG16	-	78.49% ±1.27

	True Cancer	True Inflammation	True Healthy
Predicted Cancer	13994	0	0
Predicted Inflammation	0	4032	0
Predicted Healthy	0	5	1849

In this more natural case, where correlations between subsequent frames in the input video can be exploited, our CNN architecture still outperforms other models and feature learning methods with a close to perfect performance of 99.93%. Even transfer learning of deep networks cannot compete in this section, where generalization to unseen subjects is not an issue. We conjecture that the reason is that pre-training on the large-scale data set learns a representation tailored for high generalization, which requires encoding invariances to large deformation groups into the prediction model. These invariances help to recognize natural classes, like animals and objects from daily life, even though their viewpoints and shapes might be profoundly different. It is clearly not the objective for our cross-sample use case, where generalization is less an issue than encoding extremely fine-grained similarities between samples which are very close in feature space.
Overall deep learning methods with a pre-training, the best results were obtained by the VGG16 model pre-trained on ILSVRC and fine-tuned on our target data set, where after fine-tuning a linear SVM classifier was trained on the last feature layer of the deep network. Interestingly, this performance is comparable to what was obtained in[14] for a similar colon's wall classification but on humans.

Cross-sample and cross-subject training with sample selection

We tested the performance of the handcrafted pipeline when the number of input data is limited. For this approach, images of each state are divided into training and testing sets, and then the training set is split into an increasing number of clusters based on their similarities. We stop at around 1000 clusters when a plateau of performance is reached. Then, a random image of each cluster in each state is selected to train the model, and the model is tested on the test data. Figures 4 shows the average recognition rate of the system after three trials as a function of the number of clusters, i.e., the size of the data set for the training for both cross-subject and cross-sample approaches. As visible in Fig. 4, the performance of both cross-sample and cross-subject training with sample selection overpasses the random selection of images with a gain approximately constant of 13% of recognition rate in all the range. However, at its maximum level, the performance is lower than the best performance obtained in Table 4. This approach can also be used for real-time applications as there is no need to use clustering on test data.

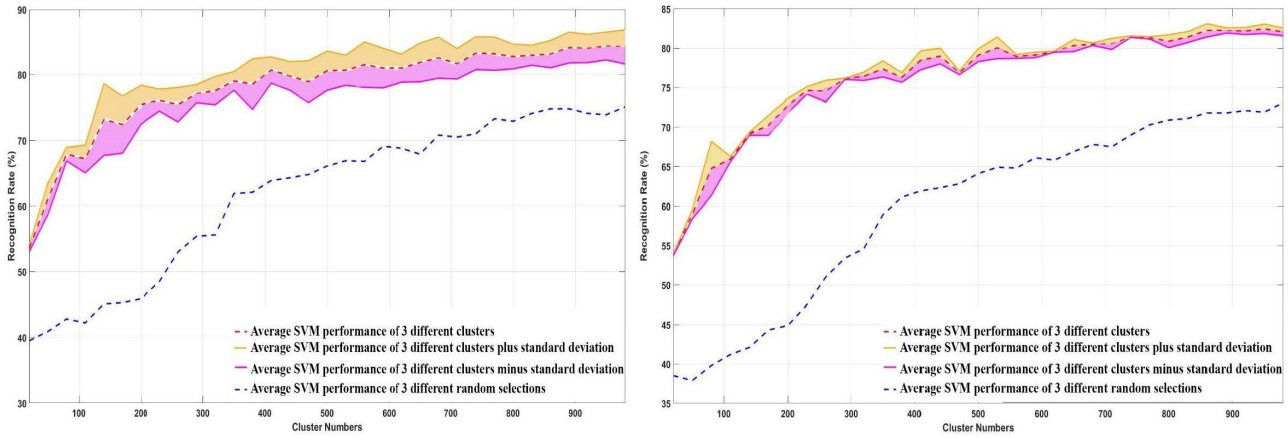


Figure 4. Average of recognition rate of cross-subject (left) and cross-sample (right) training respectively with sample selection in solid red line versus a random selection of data in dashed blue line as a function of the number of images in the training dataset. Yellow and purple lines show the average recognition rate plus and minus standard deviation respectively.

Methods

Experimental protocols and associated training strategies

Our main objective is to automate the classification process of mouse tissues into three classes, healthy, inflammation, and cancer tissues. Below, we describe two different medical use cases, where these predictions are helpful. In other words, two different approaches of splitting data into training and testing for our experiments are introduced, which refers to two different clinical problems where prediction is required on subjects or samples.

Scientific use cases

Cross-subject predictions — this use case arises when a prediction must be made on unknown subjects (unknown mice) using a model which has been created (trained) during an off-line training phase. The underlying scientific question addressed by this use case is whether locally acquired samples of tissue can be correctly classified without any additional information from the same subject. Alternatively, in other words, we would like to study whether prediction models based on machine learning can generalize to unseen subjects; it quantifies to which extent the observed diseases are generic or patient-specific.
In a real-world scenario, the corresponding prediction model is static in a sense that different predictions on new subjects will be based on the same model acquired by the medical personnel at a single instant (software updates not with standing). It means a model is trained on a given set of subjects, and will then apply it to new subjects (previously unseen). Decoupling training and prediction is the main advantage of this use case, as the prediction model does not require re-training between predictions, and results can be obtained using the same model on any new subject.

Cross-sample predictions — the second use case focuses more on individual tissue samples. This situation arises when one or more subjects are studied in detail, and a large number of tissue samples need to be classified. The underlying scientific question is, whether tissue annotation can be done semi-automatically when a large number of tissues need to be annotated from a low number of subjects. Alternatively, in other words, we would like to study whether a prediction model based on machine learning can generalize to different regions from the same or different subjects.
In a real-world scenario, the corresponding prediction model is dynamic, as (on-line) re-training is necessary for regular intervals. The medical personnel uses an application, which allows them to view tissue samples and annotate them in real-time, available in the additional information section.

The two uses cases are inherently different. Cross-subject predictions are usually more difficult, as the shift between the training data distribution and testing data distribution is generally higher, putting higher requirements on the generalization performance of the predictors. In practice, both cases can be addressed using fully supervised machine learning.

Proposed training strategies

We propose three different training strategies to address the scientific use cases described above.

Cross-subject training — this training strategy is designed to cover the cross-subject use case. The data set is split cross-subject wise, i.e., that subjects (mice) whose samples are in the training set are not present in the test set. It should be considered that the colon's wall of a subject can sometimes consist of all three labels at the same time, which means that a part of the colon's wall show cancer tissues. Another part show some inflammation tissues, and the rest can be considered as healthy tissues. Thus, it is essential to design a classifier that tries to label every image independently. Later a subject could be labeled based on the majority of labels of its images.

Cross-sample training with all samples — this strategy corresponds to the cross-sample use case. The data set is split into training and test sets by randomly sampling images of each type to be classified (health, inflammation, and cancer). In particular, this approach selects images without information on whether they are consecutive in video frames, or whether they belong to a given subject. In this strategy, images from one subject (a mouse) can be in both training and testing sets, but it does not mean that the same images are used in training and testing. As the microprobe captured images through the colon's wall of subjects, each image is taken from one specific part (tissue) of the colon's wall.

Cross-sample training with sample selection — in an alternative training strategy for the cross-sample use case, we address the fact that images correspond to video frames which are acquired in the continuity of a local probe inspection process. Therefore, consecutive images are visually similar with a high probability. This temporal correlation between frames can lead to skewed (unbalanced) data distribution and, if not dealt with, to sub-optimal performance.

We propose an unsupervised sample selection processing based on clustering. Features are extracted from each image, which includes standard deviation, mean, variance, and the skewness of the raw pixel values. The features are clustered with k-means, and a single sample is picked from each cluster for training. The rest of the images of the database are used for testing.

Features, feature learning and classification

Independently of the training strategy, we proposed two different procedures, including both feature extraction and classification methods. The first is based on handcrafted features, whereas the second resort to automatic learning of the intermediate representation.

Handcrafted features

In this methodology, we handcraft feature representations instead of learning them. Handcrafted representations have been optimized by the computer vision community over decades of research, including theoretical analysis and experiments. In our setting, we resort to the local binary patterns (LBP)[17], a state-of-the-art handcrafted descriptor which has been used in a variety of tasks in computer vision, among which are face recognition, emotion recognition, and others, see the survey in[18]. Notably, LBPs have been shown to be valuable for medical image texture analysis[19].

Under the original form of[17] and as used in this article, for a pixel positioned at the point (x,y), LBP indicates a sequential set of the binary comparison of its value with the eight neighbors. In other words, the LBP value assigned to each neighbor is either 0 or 1, if its value is smaller or greater than the pixel placed at the center of the mask, respectively. The decimal form of the resulting 8-bit word representing the LBP code can be expressed as follows:

$$LBP(x,y) = \sum_{n=0}^{7} 2^n s(i_n - i_{x,y}) \tag{1}$$

where $i_{x,y}$ corresponds to the grey value of the center pixel, and i_n denotes that of the n^{th} neighboring one. Besides, the function $s(x)$ is defined as follows:

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0\,. \end{cases} \tag{2}$$

The LBP operator remains unaffected by any monotonic gray scale transformation, which preserves the pixel intensity order in a local neighborhood. It is worth noticing that all the bits of the LBP code hold the same significance level, where two successive bit values may have different implications. The process of equation (1) is realized at the scale of a patch size of $N \times N$ pixels. The $LBP(x,y)$ of each pixel inside this patch are concatenated to create a fingerprint of the local texture around the pixel at the center of the patch. Equations (1) and (2) are applied on all patches of an image. Finally, all histogram outputs of patches (after applying LBP on them) are concatenated and considered as the feature vector of an image. This patch size N, in this study, is chosen in the order of an average size of vesicular crypts on health images. In our database, a patch size of 8×8 can almost cover a healthy vesicular crypt. At the next step, a linear SVM is applied to classify the images based on their LBP features.

Representation learning

Representation learning, or deep learning, aims at jointly learning feature representations with the required prediction models. We chose the predominant approach in computer vision, namely deep convolutional neural networks[20], which have proven to be well suited for standard tasks in the medical domain like cell segmentation[21], tumor detection, and classification[22], brain tumor segmentation[23], De-noising of Contrast-Enhanced MRI Sequences[24] and several other purposes[15]. We train two different models, one which was designed for the task and trained from scratch, and one which has been adapted from (and pre-trained on) image classification.

Training from scratch — the baseline approach resorts to a standard supervised training of the prediction model (the neural network) on the target training data corresponding to the respective training strategies described in section . No additional data sources are used. In particular, given a training set comprised of K pairs of images x_i and labels $\hat{y}_i$, we train the parameters θ of the network f using stochastic gradient descent to minimize empirical risk:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{K} \mathscr{L}(\hat{y}_i, f(x_i, \theta)) \tag{3}$$

$\mathscr{L}$ denotes the loss function, which is cross-entropy in our case. The minimization is carried out using the ADAM optimizer[25] with a learning rate of 0.001.
The architecture of our proposed architecture $f(\cdot,\cdot)$, shown in figure 5, has been optimized on a cross-validation set and is given as follows: five convolutional layers with filters of size 3×3 and respective numbers of filters 64, 128, 256, 512, 512 each followed by ReLU activations and 2×2 max pooling; a fully connected layer with 1024 units, ReLU activation and dropout (p=0.5) and a fully connected output layer for 3 classes (health, inflammation and cancer) and softmax activation.

Transfer learning — Deep learning addresses complex prediction problems through neural networks with high capacity, i.e., highly non-linear functions with a large number of parameters, whose estimation typically requires a large amount of annotated training data. If this data is not available, the trained networks tend to overfit on the training data and thus generalize poorly to unseen data.
A standard solution to this problem is transfer learning or domain adaptation. The idea is to learn high capacity models on large alternative source data sets whose content is sufficiently correlated with the target application and then transfer the learned knowledge to the target data. Various techniques have been proposed, which differ, among other in the way this transfer is performed and whether labels are available for the target data set (supervised techniques, e.g.,[26,27]) or not (unsupervised techniques, e.g.,[28]).

We perform supervised transfer using classical weight freezing and fine-tuning[26], which transfers knowledge by first solving equation 3 on the target data set, and then using the obtained parameters θ^* as initialization (starting point) for the training of the network on the target data set. The assumption is somehow grounded by the existence of standard features in images from natural scenes, which transfer well to images from other domains.

We transfer knowledge from the well-known image classification task ILSCVR 2012 (aka *ImageNet*), a dataset of roughly one million images and 1000 classes[29]. Our model architectures optimized for this task, and as described above, is very likely to underfit on this transfer learning setting. Its hyper-parameters, among which are its architecture and the number of parameters, has been optimized over a validation set, which is very much smaller than the ILVSRC data by roughly a factor of 500. Its design capacity will, therefore, tend to be much too small for the knowledge encoded in the source data (ILVSRC). For this reason, we take "classical" and well-known high-capacity models for the ILVSRC task, namely VGG16[30], DenseNet[31], and ResNet50[32]. From the pre-trained model, we remove the task-specific output layer (designed for 1000 classes) and replace it with a new layer for three classes. Among all possible combinations of freezing layers which tested, the model with freezing at the first 3 layers and fine-tuning the other layers on the validation data set returned the best performance shown in the table 3. The results of the transfer learning method with different freezing layers on our database show the transferability of features from ImageNet database in the spirit of[26].

We would liketo point out that the two different strategies (training from scratch vs. pre-training and transfer) are compared using two different model architectures. Our goal is to compare strategies, and different strategies can possibly have different optimal architectures. Network architectures need to be adapted to various parameters of the problem, namely the complexity of the task and the number of training samples. As mentioned above, in our case, there is a big difference between the small size of our dataset and the large size of typical computer vision datasets like the ImageNet/ILSVRC dataset (1M images). Therefore, this involves optimizing parameters (through SGD) as well as the hyper-parameters (through model-search). Only if both are optimized, the potentials of the two strategies are compared. In contrast, comparing two identical architectures would have been inconclusive, as one of two architectures would have been better suited to the task at hand.

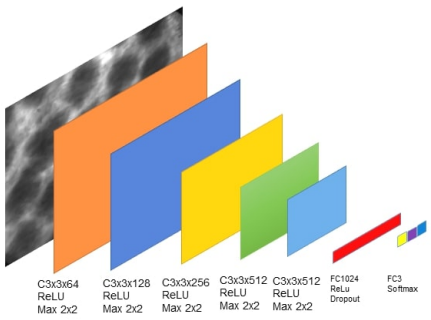


Figure 5. The proposed architecture of the deep network optimized for the task on the cross-validation set.

1 Database

The experiments involving animals were led in accordance with the rules of the University Lyon 1 Ethics Committee on animal experimentation. Animals were acclimated for two weeks prior to the experiment in the following environment: a 12-hour day/night rhythm in 300 cm^2 plastic cages (for four animals) with straw bedding, pellet food, and tap water. The temperature of each cage was monitored and kept between 19 and 21 C. To induce colitis, mice were chemically treated with a single injection of azoxymethane (AOM, intraperitoneal injection, 10mg/kg body weight) at the beginning and then, during six months, with dextran sulfate sodium in drinking water (DSS, concentration of 2%). During the experiment, a pressure sensor placed on the mouse's chest in order to monitor the respiratory index of animals. Analyzed images used in this article chosen at the extrema of the respiratory cycle, where the movements are the slowest to minimize artifacts due to these movements. Mice anesthetized with 3% isoflurane and aspiration flow set at 0.4 L / min during the induction phase. A 25 μL solution of Fluorescein Isothiocyanate FITC- Dextran 5% (Sigma Aldrich), used as a contrast agent, is injected in retro-orbital of the mouse's eye before the CEM investigation.

The anesthesia maintained during imaging with 1.4 to 1.7% isoflurane vaporization and aspiration flow set up on 0.4L/min. The endoscopic test was conducted using a mini multi-purpose rigid telescope dedicated to small animals (Karl Storz). Acquisition of images made by using a 488nm confocal endomicroscope CEM (CellVizio c, Mauna Kea Technologies) combined with a 0.95mm outer diameter Proflex MiniZ microprobe (PF-2173, Mauna Kea Technologies). The microprobe was inserted through the operating sheath of this endoscope and positioned on the mice's colon walls. During the acquisitions, the depth assessed was approximately 58 μm for a lateral resolution of 3.5 μm and a frame rate of 12 fps. The output image size is 329×326 μm^2 corresponding to a matrix of 292×290 pixels[10].

In total, 38 mice were included in the study for a total of 66788 images which have been annotated as healthy tissue images (6474 images from 9 mice), cancer tissue images (46566 images from 13 mice) or inflammation tissue images (13748 images from 16 mice) by two experts together at the same time with a pre-knowledge of mice diseases. Images were also labeled according to the mice from which they were acquired. Annotation was realized with the help of an application (available in the additional information section) especially developed for this study freely available, as pointed in the supplementary material section. It enables the classification of images according to the three classes studied in this article but also other classes of interestin biomedical studies of the colon's wall. This application is made available as supplementary material to this study. As mentioned in[5], some of the raw images do not carry any information for diagnosis. This can be due to misposition of the probe which does not receive enough signal, a decrease of the fluorescence, saturation of the imaging sensor due to too high amount of fluorescence, due to residues, due to contrast agent extravasation or presence of some light-absorbing objects within mucous film located between the probes and the tissue. To prevent the expert from spending time on annotating such non-relevant images and improve the learning process, we decided, as usually done in video endomicroscopy[33,34] to withdraw them automatically and only keep the informative frame. A simple test based on the computation of the skewness of the gray level histogram of the images demonstrated to be very efficient for this task. Images with a skewness higher than -5(as an empirical threshold) were kept. The skewness captures the dissymmetry of the histogram around its mean value. This is useful to detect saturated or underexposed images. We estimated, on some 6000 images, that this simple statistical test performs 98% of good detection for the detection of images carrying no useful diagnostic information with a false alarm of 1%. Additionally, in order to assess the influence of theses artifactual images if they would not have been removed, an additional experiment has been done on all raw data (without removing noisy data). This experiment showed a reduction of 2% (on average) on the recognition performance of each training strategy by using our proposed CNN model. This demonstrates the interest of the denoising step but also quantify the robustness of our model.

Based on the training strategies, the database was spilled into three datasets of training (for training of our model), validation (to optimize hyper-parameters), and testing (to report performance on). In the cross-subject training strategy, images of each subject (mouse) were transferred into one of the datasets of training, validation, and testing. The exact number of mice in each dataset shown in table 1. In the cross-sample training strategy, 75% of the whole database transferred to the training dataset, and the rest of the data belonged to the testing dataset. In this case, the validation dataset was extracted from the training dataset for deep learning experiments. This splitting database approach made a guaranty that the test dataset was not seen during training and validation of the model.

2 Conclusion

In this paper, we have presented three classification approaches to classify three states of health, inflammation, and cancer on mice colon's wall. Fully automated machine learning-based methods are proposed, including deep learning, transfer learning, and classical texture-based classification. Different training strategies are compared in order to find the best approach for this specific problem. The images processed in this paper were acquired in the framework of a preclinical study on colon mice. In this type of study (preclinical), the size of the database is not comparable with other domains in machine learningAs also underlined in[35] on the different types of images, we found that a custom deep learning model shows superiority over handcrafted features and well-known deep learning-based architectures. The best classification performance on this type of images are achieved with our proposed CNN model which are trained on colon's wall images.
In the cross-sample case, where generalization to unseen subjects is not an issue, Deep learning gave a performance of 99.93% of correct classification. Similar to the cross-sample, in the cross-subject approach where classification on un-seen objects is an issue, our proposed CNN method showed a performance of 98.49% of correct classification. These are usual order of magnitude of performance obtained with nowadays machine learning approaches when vast data sets are available, but this can be considered as excellent performance indeed here since we worked with the typical small data sets available in preclinical studies.

This work corresponds to the first fully automated classification algorithm for mice colon's wall images reported in the literature. Similar works were carried on the human colon's wall with the same imaging system. The comparison of the closest work[14] with our algorithm shows a comfortable margin of a 14% of accuracy. This is an interesting result which demonstrates that in the perspective of machine learning, there is no guarantee of translational research between human and animal. Also, a novel unsupervised sampling strategy based on the specific similarities of images in the acquisition of images with endomicroscopy in the colon has been designed. The interest of this sampling strategy has been demonstrated in terms of amount of data required in the training data sets to reach a plateau of performance. However, the performance of this sampling strategy is lower than brute forces classical approaches. It would be possible to improve the metric of similarity used to select the images in the training data sets automatically. This was based on first-order statistics in this study, but other approaches could be used to include more dynamical information. However, due to the multi-scale sources of temporal noise (movement of the probes[36], passing of unexpected items between probe and tissues, biological movement,etc.) it would be an open question to determine a reasonable time scale for this smoothing.

Our clustering method is somewhat related to active learning, where the agent requests feedback on data from a user. The comparison is a little bit a stretch, as no new data is collected from decisions by an agent. In our current implementation, the dataset stays stable, and only a subset is actively chosen.

However, we plan to investigate active learning as future work, where a classifier is trained on a subject followed by continued examination of the subject on new samples. Here, an agent could quickly provide decisions on i) which samples should be added to the training set, and ii) into which direction the user should emphasize its search in order to optimize performance and discovery. This leads to an exploitation/exploration trade-off known from Reinforcement learning.

Direct perspectives of other sampling strategies are possible. It would now be possible to apply the classification scheme developed here to produce a score on individual mice quantifying the number of images with the disease. Such a quantification could then be compared with clinical scores realized on other types of imaging systems in a multimodal perspective such as the one recently shown with magnetic resonance imaging[37]. Also, the machine learning approach presented with a discussion on the different training strategies could be transposed to other bioimaging problems. In confocal endomicroscopy, this includes, for instance, the characterization of other colon's diseases observed in confocal microscopy[38] or other parts of the digestive system[39] or also to other organs[40] which have received interest and could benefit from machine learning approaches to perform automated characterization of tissues.

References

1. Sirinukunwattana, K. *et al.* Gland segmentation in colon histology images: The glas challenge contest. *Med. image analysis* **35**, 489–502 (2017).

2. Brady, M., Highnam, R., Irving, B. & Schnabel, J. A. Oncological image analysis. *Med. image analysis* **33**, 7–12 (2016).
3. Becker, C., Fantini, M. & Neurath, M. High resolution colonoscopy in live mice. *Nat. protocols* **1**, 2900–2904 (2006).
4. Wang, H.-W., Willis, J., Canto, M., Sivak, M. V. & Izatt, J. A. Quantitative laser scanning confocal autofluorescence microscopy of normal, premalignant, and malignant colonic tissues. *IEEE Transactions on biomedical engineering* **46**, 1246–1252 (1999).
5. Waldner, M. J., Wirtz, S., Neufert, C., Becker, C. & Neurath, M. F. Confocal laser endomicroscopy and narrow-band imaging-aided endoscopy for in vivo imaging of colitis and colon cancer in mice. *Nat. protocols* **6**, 1471–1481 (2011).
6. Cicchi, R. *et al.* Multiphoton morpho-functional imaging of healthy colon mucosa, adenomatous polyp and adenocarcinoma. *Biomed. optics express* **4**, 1204–1213 (2013).
7. Evans, J. P. *et al.* From mice to men: Murine models of colorectal cancer for use in translational research. *Critical reviews oncology/hematology* **98**, 94–105 (2016).
8. Mielke, L., Preaudet, A., Belz, G. & Putoczki, T. Confocal laser endomicroscopy to monitor the colonic mucosa of mice. *J. immunological methods* **421**, 81–88 (2015).
9. JA Konda, V. *et al.* In vivo assessment of tumor vascularity using confocal laser endomicroscopy in murine models of colon cancer. *Curr. Angiogenesis* **2**, 67–74 (2013).
10. Bujoreanu, D. *et al.* Robust graph representation of images with underlying structural networks. application to the classification of vascular networks of mice's colon. *Pattern Recognit. Lett.* **87**, 29–37 (2017).
11. Na, K.-S. Prediction of future cognitive impairment among the community elderly: A machine-learning based approach. *Sci. reports* **9**, 3335 (2019).
12. Singh, S. P. *et al.* Machine learning based classification of cells into chronological stages using single-cell transcriptomics. *Sci. reports* **8**, 17156 (2018).
13. Min, X., Yu, B. & Wang, F. Predictive modeling of the hospital readmission risk from patients' claims data using machine learning: A case study on copd. *Sci. reports* **9**, 2362 (2019).
14. Ştefănescu, D. *et al.* Computer aided diagnosis for confocal laser endomicroscopy in advanced colorectal adenocarcinoma. *PloS one* **11**, e0154863 (2016).
15. Litjens, G. *et al.* A survey on deep learning in medical image analysis. *Med. image analysis* **42**, 60–88 (2017).
16. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626 (2017).
17. Ojala, T., Pietikainen, M. & Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis machine intelligence* **24**, 971–987 (2002).
18. Nanni, L., Lumini, A. & Brahnam, S. Survey on lbp based texture descriptors for image classification. *Expert. Syst. with Appl.* **39**, 3634–3641 (2012).
19. Nanni, L., Lumini, A. & Brahnam, S. Local binary patterns variants as texture descriptors for medical image analysis. *Artif. intelligence medicine* **49**, 117–125 (2010).
20. Ravı, D. *et al.* Deep learning for health informatics. *IEEE journal biomedical health informatics* **21**, 4–21 (2017).
21. Akram, S. U., Kannala, J., Eklund, L. & Heikkilä, J. Cell segmentation proposal network for microscopy image analysis. In *Deep Learning and Data Labeling for Medical Applications*, 21–29 (Springer, 2016).
22. Akselrod-Ballin, A. *et al.* A region based convolutional network for tumor detection and classification in breast mammography. In *Deep Learning and Data Labeling for Medical Applications*, 197–205 (Springer, 2016).
23. Zhao, X. *et al.* A deep learning model integrating fcnns and crfs for brain tumor segmentation. *Med. image analysis* **43**, 98–111 (2018).
24. Benou, A., Veksler, R., Friedman, A. & Raviv, T. R. De-noising of contrast-enhanced mri sequences by an ensemble of expert deep neural networks. In *Deep Learning and Data Labeling for Medical Applications*, 95–110 (Springer, 2016).
25. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. In *ICML* (2015).
26. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*, 3320–3328 (Curran Associates, Inc., 2014).
27. Douarre, C., Schielein, R., Frindel, C., Gerth, S. & Rousseau, D. Transfer learning from synthetic data applied to soil–root segmentation in x-ray tomography images. *J. Imaging* **4**, 65 (2018).

28. Ganin, Y. & Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *ICML* (2015).

29. Russakovsky, O. *et al.* ImageNet Large Scale Visual Recognition Challenge. *IJCV* **115**, 211–252 (2015).

30. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR* (2015).

31. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, vol. 1, 3 (2017).

32. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).

33. Oh, J. *et al.* Informative frame classification for endoscopy video. *Med. Image Analysis* **11**, 110–127 (2007).

34. Bashar, M. K., Kitasaka, T., Suenaga, Y., Mekada, Y. & Mori, K. Automatic detection of informative frames from wireless capsule endoscopy images. *Med. Image Analysis* **14**, 449–470 (2010).

35. Murthy, V. N. *et al.* Cascaded deep decision networks for classification of endoscopic images. In *Medical Imaging 2017: Image Processing*, vol. 10133, 101332B (International Society for Optics and Photonics, 2017).

36. Latt, W. T. *et al.* A hand-held instrument to maintain steady tissue contact during probe-based confocal laser endomicroscopy. *IEEE transactions on biomedical engineering* **58**, 2694–2703 (2011).

37. Dorez, H. *et al.* Endoluminal high-resolution mr imaging protocol for colon walls analysis in a mouse model of colitis. *Magn. Reson. Mater. Physics, Biol. Medicine* **29**, 657–669 (2016).

38. Neumann, H. *et al.* Confocal laser endomicroscopy for in vivo diagnosis of clostridium difficile associated colitis—a pilot study. *PLoS One* **8**, e58753 (2013).

39. Liu, J. *et al.* Learning curve and interobserver agreement of confocal laser endomicroscopy for detecting precancerous or early-stage esophageal squamous cancer. *PloS one* **9**, e99089 (2014).

40. Foersch, S. *et al.* Confocal laser endomicroscopy for diagnosis and histomorphologic imaging of brain tumors in vivo. *PLoS One* **7**, e41760 (2012).

Acknowledgment

This work was supported by the LABEX PRIMES (ANR-11-LABX- 0063) of Université de Lyon, within the program Investissements d'Avenir (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR) as well as DORA plus (Estonian government programme).

Author contributions statement

Conceptualization, P.R. and C.W. and D.R; Data curation, H.D. and R.S. and D.M.; Formal analysis, P.R. and D.R.; Methodology, D.R.; Software, P.R. and S.S.; Supervision, D.R.; Validation, P.R. and S.S. and D.R.; Visualization, P.R.; Writing – original draft, P.R. and D.R.. All authors reviewed the manuscript.

Additional information

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this article.

Research involving animals: All applicable international, national, and/or institutional guidelines for the care and use of animals were followed. All procedures performed in studies involving animals were in accordance with the ethical standards of the institution or practice at which the studies were conducted.

Ethical standards: This study was approved by the institutional review board of the Université Claude Bernard Lyon 1 (reference number: DR2014-62-v1) and complied with ethics committee standards.

Annotating software: The annotating software tool has been specially developed for this study but is applicable to any video endoscopy annotation for cancer. It is freely available at https://uabox.univ-angers.fr/index.php/s/AZ2IZl6LDYRcd8P together with a demo video and some data sample.

Titre : Contributions à l'imagerie à bas coût et à l'apprentissage automatique pour le phénotypage des plantes.

Mot clés : imagerie, apprentissage machine, phénotypage

Résumé : Dans cette thèse, nous étudions les possibilités de réaliser une imagerie à haut débit pour le phénotypage végétal à faible coût sur un ensemble de questions biologiques. Nos contributions peuvent être organisées en deux parties. La première partie se concentre sur la façon de réduire le coût du phénotypage végétal au niveau du capteur. Dans cette section, nous montrons l'utilisation novatrice des mini-ordinateurs, associés aux caméras RVB et/ou LiDAR, pour surveiller les plantes à partir de la vue de dessus en tant qu'individus, ou au niveau de la canopé. Avec un accès plus pratique aux systèmes d'imagerie, le goulot d'étranglement actuel du phénotypage végétal correspond désormais au développement d'algorithmes de traitement d'image optimisés. La deuxième partie traite de cette question et se concentre sur la réduction du coût de calcul et du temps requis pour la création de la vérité-terrain associée aux images à traiter. Nous avons étudié la valeur de la transformation scatter, qui est une architecture de réseaux profonds ne nécessitant pas de ressources informatiques massives ou de grands ensembles de données annotés. Nous avons également étudié la possibilité d'effectuer des annotations d'images automatisées avec un apprentissage automatique non supervisé dans des séquences d'images. Nous avons démontré, la possibilité d'accélérer l'annotation avec des outils ergonomiques basés sur la capture de la direction du regard de l'annotateur. Enfin, nous avons démontré la possibilité d'accélérer l'annotation en utilisant des données synthétiques annotées automatiquement.

Title: Contributions to low-cost imaging and machine learning for plant phenotyping.

Keywords: Low-cost plant Phenotyping, imaging, Machine Learning.

Abstract: In this thesis, we investigate the possibilities of performing high-throughput imaging for plant phenotyping at low cost on a set of biological questions. Our contributions can be organized into two parts. The first part focuses on how to reduce the cost of plant phenotyping at the sensor level. In this section, we show the innovative use of mini-computers, associated with RGB and/or LiDAR cameras, to monitor plants from the top view as individuals , or at a canopy level . With more convenient access to imaging systems, the current bottleneck in plant phenotyping now corresponds to the development of optimized image processing algorithms. The second part addresses this issue and focuses on reducing the computational cost and the time required for the creation of ground-truth associated with the images to be processed. We have investigated the value of the scattering transform, which is a deep architecture without the need for massive computational resources or large annotated datasets. We have also investigated the possibility of performing automated image annotation with unsupervised machine learning in sequences of images . We have demonstrated, the possibility to speed up annotation with ergonomic tools based on the capture of the annotator's gazing direction . Last, we have demonstrated the possibility to speed up annotation by using synthetic data automatically annotated.