U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

MSc

Automatic processing of images of chromotropic traps for identification and quantification of Trioza erytreae and Scaphoideus titanus

Beatriz Lobo da Silva Pinto Bessa

Dissertação de Mestrado apresentada à
Faculdade de Ciências da Universidade do Porto em
Bioinformática e Biologia Computacional

2021

MSc
2.º CICLO
FCUP
2021

U.PORTO

Automatic processing of images of chromotropic traps for identification and quantification of Trioza erytreae and Scaphoideus titanus

Beatriz Lobo da Silva Pinto Bessa

FC

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

MSc

Automatic processing of images of chromotropic traps for identification and quantification of Trioza erytreae and Scaphoideus titanus

Beatriz Lobo da Silva Pinto Bessa
Mestrado em Bioinformática e Biologia Computacional
Departamento de Biologia
2021

Orientador
André Marçal, Professor Auxiliar, FCUP

Coorientador
Mário Cunha, Professor Associado, FCUP

Agradecimentos

Gostaria de expressar a minha imensa gratidão a todos aqueles que, de alguma forma, contribuíram, direta ou indiretamente, para a realização desta dissertação.

Primeiramente, gostaria de agradecer ao Prof. André Marçal e ao Prof. Mário Cunha, da Faculdade de Ciências da Universidade do Porto, pela orientação, apoio e por permitirem liberdade para expressar as minhas ideias.

Gostaria também de agradecer ao Engenheiro Seabra pela sugestão do tema, disponibilidade, e por me oferecer todos os contactos necessários para obtenção de dados.

À Engenheira Amália Xavier, um sincero e imenso agradecimento por todas as horas dedicadas a ajudar-me, por todo o apoio, pela simpatia, e por ter acreditado em mim, e neste projeto.

Ao Engenheiro Artur Santos gostaria de agradecer pela prontidão em ajudar-me, por me deixar acompanhá-lo na recolha das armadilhas, e por me mostrar toda a beleza da região da Régua e Lamego.

À minha família, por todo o amor incondicional, e pelas palavras de motivação em fases difíceis. Gostaria de agradecer especialmente ao meu irmão, por ser o meu eterno parceiro.

A todos os meus colegas, que contribuíram para que toda esta experiência fosse exponencialmente mais divertida. Tive a oportunidade de conhecer algumas das pessoas mais inteligentes e incríveis durante esta jornada e, por isso, estou muito grata.

A todos os meus amigos, agradeço simplesmente por estarem presentes na minha vida.

À “cave do césar”, pelo espírito de camaradagem, cooperação e boa disposição.

Ao Francisco, por todo o amor e constante incentivo.

Resumo

Trioza erytreae (Del Guercio) é vetor da doença huanglongbing, ou enverdecimento dos citrinos, e *Scaphoideus Titanus* Ball é vetor da doença da vinha conhecida como flavescência dourada. Ambas as formas adultas destes insetos são atraídas pela cor amarela, portanto, armadilhas cromotrópicas, também conhecidas como placas amarelas ou placas anti pragas, são utilizadas para detetar a presença destas pestes, que deve ser confirmada por um entomologista. Seguindo os recentes avanços em visão computacional e *deep learning*, este trabalho combina técnicas de processamento de imagem e dois detetores de objetos populares – Faster R-CNN e YOLOv4 – para deteção de ambos os insectos em imagens de armadilhas cromotrópicas antes e depois de serem processadas. Ambos os modelos tiveram um baixo desempenho, particularmente o Faster R-CNN, e modelos treinados com as imagens processadas obtiveram scores ainda mais baixos, realçando a necessidade de rever o método de processamento utilizado.

Abstract

Trioza erytreae (Del Guercio) is vector of the huanglongbing or citrus greening disease, and *Scaphoideus Titanus* Ball, is vector of a vineyard disease known as flavescence dorée. Both adult forms of these insects are attracted to the yellow color, therefore, chromotropic traps, commonly known as yellow sticky traps, have been widely used to detect the presence of these pests, which should be confirmed by an entomologist. Following the recent advancements in computer vision and deep learning, this work combines image processing techniques and two popular object detectors – Faster R-CNN and YOLOv4 – for detection of both insects in images of chromotropic traps before and after they were processed. Both models underperformed, particularly the Faster R-CNN, and models trained with processed data obtained even lower scores, emphasizing the need for revising the processing method used.

Keywords: *Trioza erytreae*, *Scaphoideus Titanus*, Image processing, Computer vision, Choromotropic traps, Neural networks, Deep learning, Entomology, Object detection, YOLOv4, Faster R-CNN.

Index

Figure and table Index

Abbreviation list

ANN	Artificial Neural Networks
AP	Average Precision
BoF	Bag of Freebies
BoS	Bag of Specials
CIoU	Complete Intersection over Union
CmBN	Cross mini–Batch Normalization
CNN	Convolutional Neural Networks
CSP	Cross Stage Partial connections
CV	Computer vision
DGAV	Direção-Geral de Alimentação e Veterinária
DIoU	Distance Intersection over Union
DL	Deep learning
DRAP	Direção Regional de Agricultura e Pescas
DRAPN	Direção Regional de Agricultura e Pescas do Norte
FCL	Fully Connected Layer
FD	Flavescence dorée
FPN	Feature Pyramid Network
FRCNN	Faster R-CNN
HLB	Huanglongbing disease
IoU	Intersection over Union
mAP	Mean Average precision
MiWRC	Multi-input Weighted Residual Connections
PAN	Path Aggregation Network
ReLU	Rectified Linear Unit
RoI	Regions of Interest
RPN	Region Proposal Network
SAM	Spatial Attention Module
SAT	Self-Adversarial Training
SPP	Spatial Pyramid Pooling
SS	Selective Search
ST	Scaphoideus titanus
TE	Trioza erytreae
ZIP	Zonas de Intervenção Prioritárias

1. Introduction

Trioza erytreae Del Guercio *(*TE), or African citrus psyllid, and *Scaphoideus Titanus* Ball (ST), or American grapevine leafhopper, are vectors of some of the most devastating citrus and vineyard diseases, respectively. Although both insects are not originally from Europe, ST was first reported in France in 1958 due to the import of American rootstocks, and TE in Madeira in 1994 and consequently in continental Portugal and Spain in 2014 (DRAPCentro, 2017; EPPO Global Database, 1995; Pérez-Otero et al., 2015). Since ST has been spreading since the 50's *Flavescence dorée* (FV), caused by phytoplasmas, has been disseminated across Europe, while Huanglongbing disease (HLB), vectored by TE and caused by *Candidatus Liberibacter* spp., has not been reported in Europe so far. However, due to the disruptive nature of this disease, avoiding its spreading once it is identified, is of extreme importance. Both pathogens belong to the *Candidatus* phyla, a taxonomic status for prokaryotic cells that are characterized but uncultured, which may difficult their study and understanding (Murray and Stackebrandt, 1995; Stackebrandt et al., 2002).

Chromotropic traps, also known as yellow sticky traps, are commonly used to capture insects on the field. Studies have reported that both TE and ST are attracted to the yellow color and, therefore, these traps prove to be an effective way to assess their presence on the area (Mazzoni et al., 2011; Samways, 1987). However, there have been reports suggesting that male ST were more attracted to red traps (Lessio and Alma, 2004). The phototaxic preference of male ST should be better comprehended since

Fig. 1 - Example of chromotropic trap in a grapevine in Lamego.

some studies suggest that males have higher interplant movement due to the constant movement to find mates, and therefore are responsible for a higher infection rate (Lessio and Alma, 2004). Regardless, yellow traps were retrieved from different regions to identify TE and ST.

Parallelly to technological developments, computer vision (CV) has been replacing manual labor and inspection in agriculture. There have been several studies that use vision systems in precision farming (Hemming and Rath, 2001; Patrício and Rieder, 2018), fruits and vegetables quality evaluation (Bhargava and Bansal, 2021), crop health (Chouhan et al., 2020) and many more. With the recent advances in CV, sensors and cameras, it is also becoming increasingly efficient to monitor insect populations in a less-invasive and more cost-effective way (Høye et al., 2021). There have been several studies that use CV approaches to try to solve entomological problems. Ding and Taylor (2016) used a sliding window detection pipeline with a convolutional neural network (CNN) to detect codling moths achieving a precision-rate of 93.1%; Hong et al. (2020) compared seven object detectors for moth detection; Kim et al. (2019) used a fully convolutional network to detect mosquitos and obtained an accuracy of 84%; Thenmozhi and Reddy (2019) proposed a CNN model and compared it to other pre-trained DL architectures such as AlexNet, ResNet, GoogLeNet and VGGNet for insect classification on 3 different datasets. The highest accuracy reported was of 96.75% for the NBAIR insect dataset, which has 40 classes.

Although often confused with image classification, object detection can identify different objects and their location (Lamba 2019). Object detection complexity increases when the problem concerns several objects with different sizes – since the variability in the output of network is not constant, the problem cannot be solved by a typical fully connected network. Moreover, simply cropping out different regions of an image and classify them using a CNN, is not efficient – it would require cropping out sub-images with many different sizes and aspect ratios, becoming computationally expensive. Hence, the creation of object detectors (Gandhi, 2018). Two-stage detectors, such as the R-CNN family (Girshick et al. 2014), generate regions of interest and pass them on to a network for object classification and bounding-box regression. One-stage detector, such as YOLO (Redmon et al. 2016), divide the image into different cells, draw a class probability map from them and, simultaneous, extract bounding-boxes for that cell. One-stage-detector usually have lower accuracy than two-stage detectors, but are faster, so are usually implemented in real-time object detection (Soviany and Ionescu 2018).

The presented work focused on the identification of TE and ST in images of chromotropic traps. The images were segmented to obtain the entire trap, and then an algorithm for the correction of distortions was implemented. The influence of characteristics such as lighting and distortions of the trap in this process was also examined. Afterwards, the images were fed to a Faster R-CNN and a YOLOv4 model. To grasp the influence of segmentation and geometric transformation procedures, these models were also trained on the original images.

2. Literature Review

2.1. Insect vectors

2.1.1. Trioza Erytreae

HLB is caused by the gram-negative (bacteria with a cell wall) *Candidatus Liberibacter* species - *Candidatus Liberibacter asiaticus*, *Candidatus Liberibacter americanos*, which are vectored by the Asian Citrus Psyllid, *Diaphorina citri*, and *Candidatus Liberibacter africanus*, vectored by the African Citrus Psyllid, TE. (Bové, 2006; Wang, 2019). Based purely on symptoms produced, the disease caused by the three agents is indistinguishable from each other (Aglave, 2019). Since these agents belong to the *Candidatus* phyla, a taxonomic status for prokaryotic cells that are yet to be cultured *in vitro*, the pathogenesis of HLB is challenging to understand (Murray and Stackebrandt, 1995; Stackebrandt et al., 2002; Wang, 2019). Although this disease is not completely understood, studies have shown that infected trees have partial or total phloem collapse (Folimonova and Achor, 2010).

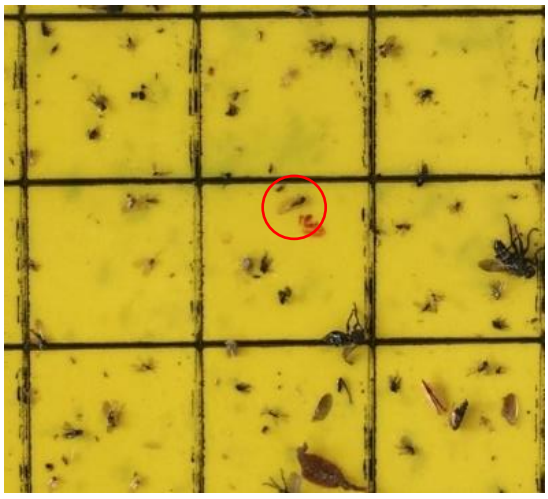


Fig. 2 - Trioza erytreae on a chromotropic trap.

Symptoms are variable and can be numerous. Also, the timeframe from infection to the manifestation of symptoms ranges from months to years depending on the cultivar, tree age and health, environmental conditions and time of the year (Aglave, 2019). Before any of the foliar symptoms caused by phloem plugging, these trees present root infection, suggesting replication at the roots prior to spreading to rest of the canopy (Johnson et al., 2014). Infected trees can present blotchy or completely yellow leaves (Fig. 3a), similar do zinc deficiencies, enlarged veins, and produce small, asymmetrical, bitter fruits with mottled yellow/green coloration and dark aborted seeds, as presented in Fig. 3b (Aglave, 2019; Halbert and Manjunath, 2004). Due to the disruptive nature of this disease and no effective way to combat it so far, it greatly impacts citrus production. In the United States, from 2007-2008 to 2017-2018, HLB was responsible for a decrease in orange production for processing of 72.2%, and a decrease in production for fresh fruit market of 20.5%. Since its detection in the US, the price of a box of oranges increase 3.2 times (Dala-Paula et al., 2019) .

HLB has been reported in China in 1919, in 1937 in South Africa and in the Philippines in the 1960's. Since then, it has been reported in 50 countries (Dala-Paula et al., 2019). In Europe, HLB is yet to be reported but its vector, TE, was firstly reported in Madeira in 1994 (EPPO Global Database, 1995) and in the Canary Islands in 2002

a)

b)

Fig. 3 - Symptoms of HLB. a) Leaves from an infected plant [Source: Sprague, 2019.] b) Fruits from an infected tree on the center, compared to healthy ones [Source: Bové, 2006].

(Hernández, 2003). However, the first appearance in a European continental area was in 2014 in the northern-west part of the Iberic Peninsula (DRAPCentro, 2017).

Although HLB is extremely destructive to citrus, the very presence of TE can disrupt the growth of the plant. TE is closely associated with plants of the Rutaceae family (Cocuzza et al., 2017). Its nymphs settle and feed on the underside of tender leaves, causing them to become chlorotic and slightly curled, as seen in Fig. 4, which can eventually cause more than 90% of young plants to die if insecticide is not used (Cocuzza et al., 2017; Tamesse and Messi, 2002). Additionally, severe infestation causes an abundance of honeydew excreted by nymphs, facilitating the development of fungi and can attract ants (van den Berg et al., 1991).

TE is easily identified by its forewing. It is 2.79–3.09 times longer than wide, with elongate oval and narrowing to a rounded rectangular apex. This specie is very small even when compared to other insects, with female body length usually around 2.2 mm and males around 1.8 mm (Fig. 2 and Fig. 5) (Aidoo et al., 2019).

Currently, Portugal's plant protection measures include the establishment of a delimiter zone. In case of official confirmation of the presence of TE, either by morphological identification or by molecular analysis of insects captured in chromotropic traps or in plant material, this area is immediately defined, formed by the parish where the insects were detected, considered an infested area, and by a surrounding buffer zone of 3 km radius, established from the limits of infested regions. Any owner or any professional operator who produces or sells host plant material and who is aware or

Fig. 4 – Leaves deformed by Trioza erytreae [Courtesy of Carlos Alberto Coutinho Conceição.]

suspects the presence of TE, must immediately inform the phytosanitary inspection services of the respective DRAP (Direção Regional de Agricultura e Pescas) or DGAV (Direção-Geral de Alimentação e Veterinária).

Producers and suppliers of citrus, whose place of activity is covered by the defined demarcated zone may only sell or distribute citrus if certain following conditions are met, such as:

- Production or maintenance of vegetables, for at least one year, in insect-proof locations that prevent the introduction of TE, previously approved and registered, verified by DRAP, and subject to at least two official inspections annually during the production cycle.
- The transport of plants within the delimited area, can only be from places that meet the characteristics referred above and should be fully wrapped in plastic film or other material that prevents direct contact with the.
- Transport, reception and shipment of vegetables in closed containers or packages;

Additionally, non-professionals that have host plants also have measures to comply with, such as carry out phytosanitary treatments to these vegetables with authorized plant protection products [1].

a) ***b)***

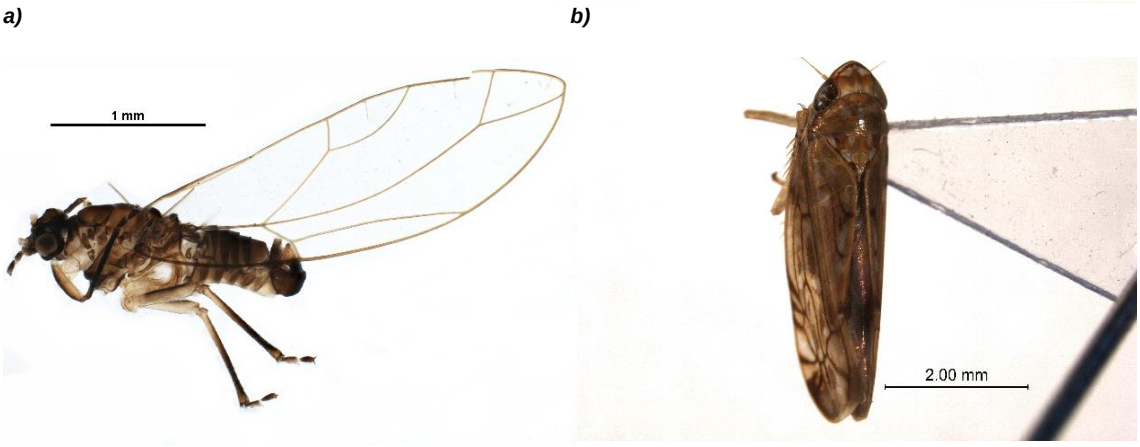


Fig. 5 -a) Trioza erytreae [Source: CBG Photography Group, 2016]. b) Scaphoideus titanus [Source: CNC/BIO Photography Group, 2012].

[1] Portaria nº 142/2020 de 17 de Junho. *Diário da República nº 116/2020 – I Série*. Ministério da Agricultura. Lisboa.

2.1.2. Scaphoideus Titanus

FD is one of several grapevine yellows diseases caused by phytoplasmas, cell-wall lacking bacteria that inhabit sieve elements and are vectored by insects that feed on the plants phloem (Dermastia et al., 2017). FD is associated with phytoplasmas part of the 16Sr-V group, specifically with 16SrV-C and 16SrV-D phytoplasmas, which are notably smaller than other bacterial plant pathogens, making them hard do visualize under a light microscope, and are yet to be cultivated outside their hosts (Chen et al., 2012; Hogenhout, 2009; Martini et al., 1999).

In Portugal, ST was detected in 2000 but it was only in 2007 that the first case of FD was reported on the country (Aguin-Pombo et al., 2020; Quartau et al., 2001). However, in other European countries the first reports of this disease go further back. For example, the first epidemy of FD in France was in the 1950's (Caudwell, 1990). Currently, FD is present in 16 European countries, making the research on this disease in Europe, when compared to HLB, vaster (EPPO Global Database, 2021).

Vineyards can differ in their susceptibility to FD, but it is often epidemic, as the proportion of affected grapevines within a vineyard can reach 95% (Dermastia et al., 2017). Although the mechanisms through which phytoplasmas affect plant development are unclear, it is believed that these parasites can produce effectors that regulate characteristic targets in their hosts (Chen et al., 2012; Sugio et al., 2011). Yellowing of leaves on white cultivars, reddening on red cultivars, with downward curling, drooped canes due to the lack of lignification in the new shoots, and death of inflorescences and berries, are some of the symptoms of this disease (Fig. 6) (Chuche and Thiéry, 2014). This disease greatly affects production levels. In a study made from 1999 to 2003 to evaluate the productivity of FD infected vineyards in Italy, symptomatic vines always showed lower yields, followed by recovered plants. In this study, recovered plants always

a) ***b)***

Fig. 6 – a) Example of FD in a white grapevine. b) Example of FD in a red grapevine [Source: Del Serrone, 2007]

showed approximately 80% more yield than plants with FD, but never as much as healthy ones (Morone et al., 2007).

Contrarily to TE and citrus, ST itself does not present a threat to grapevines, the main concern regards the dissemination of phytoplasmas, particularly adult specimens since nymphs usually stay on the plant where they hatch (Maixner, 1993). Some studies even suggest that males tend to disperse more than females, possibly to find mates, hence, may infect more plants (Lessio and Alma, 2004).

Morphologically, ST is easier to identify by the naked eye, comparatively to TE, mainly due to difference in size. A study in Portugal in 2001 showed that male length was from 4.49-5.02 mm, and female 5.15-5.48 mm (Quartau et al., 2001). Fig. 5 displays a side-to-side comparison of both species. ST also has very distinct bands, females have three brown transverse bands at the vertex level while males only have one (Tramontini et al., 2020).

The current strategies to combat it include delimiting ZIP areas (Zonas de Intervenção Prioritárias), territory from localities where strains contaminated with FD are detected, and by the respective neighboring and non-border localities that are covered by the perimeter to be defined in the information obtained through the Vine and Wine Information System. The updated list of ZIP areas is published every year in the DGAV website. The owners, of Vitis spp. located in the parishes where the ST is present, must annually carry out insecticide treatments, with plant protection products authorized by the DGAV, in appropriate timing, and the number of treatments can range from one to three. If FD is confirmed, the destruction by fire of all contaminated strains located within a 1000 m perimeter is mandatory for that year and following years. If the number of contaminated strains in a parcel is greater than 20% of the total number of strains in that parcel, the destruction of the entire vine parcel is mandatory. Regarding nurseries, their

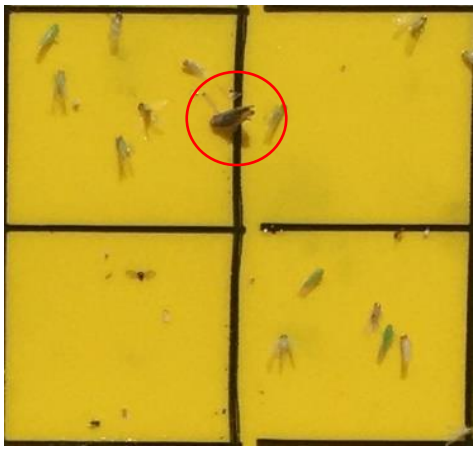


Fig. 7 - Scaphoideus Titanus in a chromotropic trap.

planting without physical protection against ST is prohibited, at less than 300 m from a contaminated parcel. From 300m to 1000m, or if there is any risk of contamination, the material retrieved from the nursery, must undergo hot water treatment [2].

2.2. Deep Learning

Deep learning (DL) is a subset of machine learning that is becoming increasingly more relevant in modern problems. An overview of several definitions in order to reach a unified description of the term has been attempted by Zhang et al. (2018), who finally described it as *" a process not only to learn the relation among two or more variables but also the knowledge that governs the relation as well as the knowledge that makes sense of the relation".*

In 1943 McCulloch and Pitts proposed the MCP model, a highly simplified computational model that mimicked the brains neuron, which would make an important contribution to the current Artificial Neural Networks (ANN) (Mcculloch and Pitts, 1943). Several breakthroughs have been made following the MCP, like the perceptron by Rosenblatt in 1958 and the neocognitron by Fukushima in 1980, which was capable of pattern recognition and introduced for the first time convolutional and downsampling layers (Fukushima, 1980; Rosenblatt, 1958). However, due to the limited computational power for training, DL was not widely used. It was only decades later that some of the significant achievements were met in CV. Following several past accomplishments, AlexNet, a CNN designed by Krizhevsky et al. (2012), won ImageNet Large Scale Visual Recognition Challenge, achieving an unprecedented reduction in the error rate of 18% and it has been recognized ever since as one of the main breakthroughs in CV history.

The advances in the last decade that boosted DL can be mainly attributed to the appearance of publicly available, high-quality, large, labelled datasets, as well as the enabling of GPU-based training, which significantly reduced training speed (Voulodimos et al., 2018). Given the necessary power, DL is more efficient than machine learning, since it does not require to manually perform feature extraction and engineering, which is done while training. However, it requires a larger amount of data (Kim, 2017). In recent years, DL has been applied to a variety of research fields. In CV, DL has made an enormous contribution to problems such as medical imaging (Ker et al., 2018; Litjens et al., 2017), autonomous vehicle control (Kuutti et al., 2021), face recognition (Masi et al., 2018), and many more (Sejnowski, 2018).

[2] Portaria n.º 165/2013 de 26 de Abril. *Diário da república nº 81/2013 – I Série.* Ministério da Agricultura, do Mar, do Ambiente e do Ordenamento do Território. Lisboa.

2.2.1. Convolutional Neural Networks

In 1962 David Hubel and Torsten Wiesel described the concept of “simple” and “complex” cells, after studying how cat’s visual cortex responded to stimulus. Both types responded to oriented slits of light, but simple fields have distinct excitatory and inhibitory divisions, whereas complex cells did not have a clear difference between these regions but encompassed a wider area within a specific orientation. They argued that the properties of complex cells could result from combining input from similarly oriented simple cells (Hubel and Wiesel, 1962, 1959). The idea of simple and complex cells would later inspire Dr. Kunihiko Fukushima to create the neocognitron – a network that would mimic this system and would be composed of S-cells and C-cells. Like Hubel and Wiesel’s cells, the higher the complexity, higher the tendency to respond selectively to more complicated features of the stimulus pattern, and, simultaneously, show less sensitivity to shifts in the position of the pattern and have a larger receptive field. Hence, the total number of cells per layer decreases with the depth in the network. In the last layer, the receptive field of each C-cell becomes large enough to cover the whole area of input layer. Following the neocognitron there were several achievements from CNNs throughout the years, including the previously mentioned AlexNet.

CNNs are now widely used in CV. The distinctive trait CNNs are, as the name implies, the convolutions, which can be seen as sliding windows passing over the image, creating layers of features (Sejnowski, 2018). Specifically, these filters (convolutional filters or kernels) are two-dimensional matrices and, depending on what filters are, extract different features from the original image, creating feature maps. As mentioned, the work of Hubel and Wiesel suggested that the brain does not interpret an image directly, there seems to be a sequential process and starts by detecting low-level elements such as lines. The filters in the first layer are analogous to what they called “simple cells” in the primary visual cortex (Hubel and Wiesel, 1962; Sejnowski, 2018). As the depth in the network increases, so does the complexity of the features extracted from each convolution (Albawi et al., 2017; Kim, 2017).

2.2.2. R-CNN, Fast R-CNN and Faster R-CNN

In 2014, Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik released the R-CNN: Region proposal with CNN features (Girshick et al., 2014). This model used an algorithm known as Selective Search (SS), which combines exhaustive search and segmentation based on pixel similarity, to propose regions of interest (Kar, 2020; Uijlings et al., 2013). From an image, 2000 candidate region proposals were fit into a square and processed through a CNN that would act as a feature extractor, and the obtained features were fed into a Support vector machine to classify the presence of the object within the region. This new CNN defeated the previous best result on PASCAL VOC 2012 by an improvement of 30% (Girshick et al., 2014). However, this method was very slow, it took a long time to process training imagens and around 49 seconds for each test image, if included the regional proposal methods. Additionally, SS is a fixed algorithm, it cannot learn. Ross Girshick improved his own work and released Fast R-CNN in 2015 (Girshick, 2015). Instead of cropping out and simply feeding 2000 region proposals to the CNN, the input image would be directly fed to the CNN to generate a convolutional feature map corresponding to the whole image. Afterwards, using a proposal method such as SS on the original image, the regions proposed would be projected onto the previously created feature map, generating regions of interest (RoI) of the feature map, and not the image. Fully connected layers (FCLs) downstream impose a fixed aspect ratio, so these RoI were processed through a pooling layer. Finally, the FCLs would predict a classification score and return the bounding boxes with offsets. The ability to reuse the feature map of the whole image instead of processing 2000 different images through the network drastically improved speed. Training time in R-CNN improved from 84 hours to 8.75 hours in Fast R-CNN, and test time from 49 seconds to 2.3 seconds, including region proposal. However, these 2.3 had potential to be improved,

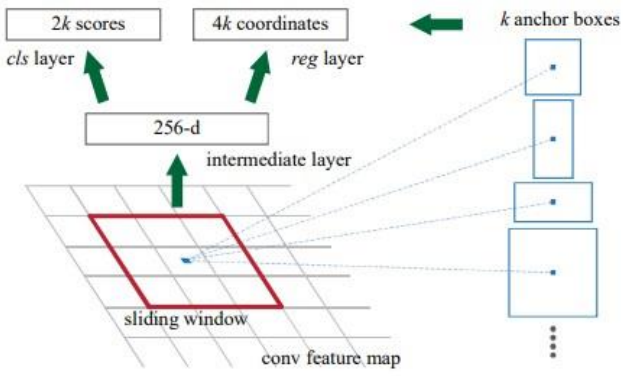


Fig. 8 – Region proposal network [Source: Ren et al., 2016].

given that if the regional proposal methods were not included, the network would take 0.32 to provide classification and bounding boxes (Girshick, 2015; Li et al., 2017). This obstacle would later be overcome by the Faster R-CNN (FRCNN). This network would completely eliminate the need for methods as SS. Instead, FRCNN has two modules. It firstly trains a CNN to generate region proposals known as the region proposal network (RPN), and then uses the detector with the proposed regions. The implementation of RPN would bring down the computational time from 2s to 10ms per image. (Ren et al., 2016; Shilpa Ananth, 2019).

RPN is a fully convolutional network designed to propose regions within a diverse range of scales and aspect ratios, as opposed to previous methods. This is achieved by a pyramid of anchors, which is more cost-efficient than using pyramids of images or pyramids of filters (images or filters at different scales). An anchor is the center point of a sliding window and is associated with different scales and aspect ratios. RPN takes as input an $n \times n$ spatial window of the input convolutional feature map (Fig. 8). Each of these sliding windows is mapped to a lower-dimensional feature which is then fed into two FCLs—a regression layer and a classification layer. At each sliding window location, k region proposals are predicted. The regression layer outputs $4k$ coordinates of k boxes, the classification layer outputs $2k$ scores that estimate the probability of each proposal have an object, or not. Contrarily to SS, the RPN can be trained. However, since the RPN and the Fast R-CNN are unified into one, they cannot be trained as individual models (Ren et al., 2016).

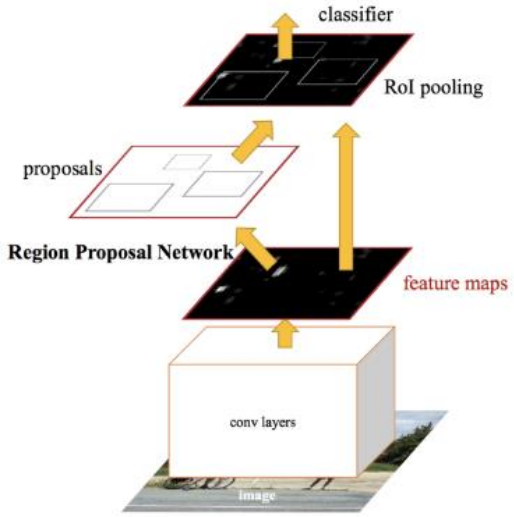


Fig. 9 - Faster R-CNN [Source: Ren et al., 2016].

2.2.2.1. Backbone

Originally, the FRCNN was implemented with a VGG-16 backbone, or feature extractor. At the time VGG-16 was certainly a major improvement from the previous methods, however, there are currently better options to choose from. During this work, the network used as backbone for the FRCNN was a ResNet, which has better performance, is faster, and has less parameters which means that is requires less memory than VGG, and was actually created by three of the four authors that developed the FRCNN (He et al., 2016).

After AlexNet, research was still very much focused on creating deeper and deeper networks. However, as He et al. (2016) noted, this does not always result in better performance, in some cases these larger networks would underperform comparatively to their shallower counterparts. The authors suggested that a deeper network should at least perform as well, they should not have a higher training error, since the deeper network could learn like the shallower one in the first layers, and simply apply identity functions for the rest of the layers, but these identity functions seemed hard for the

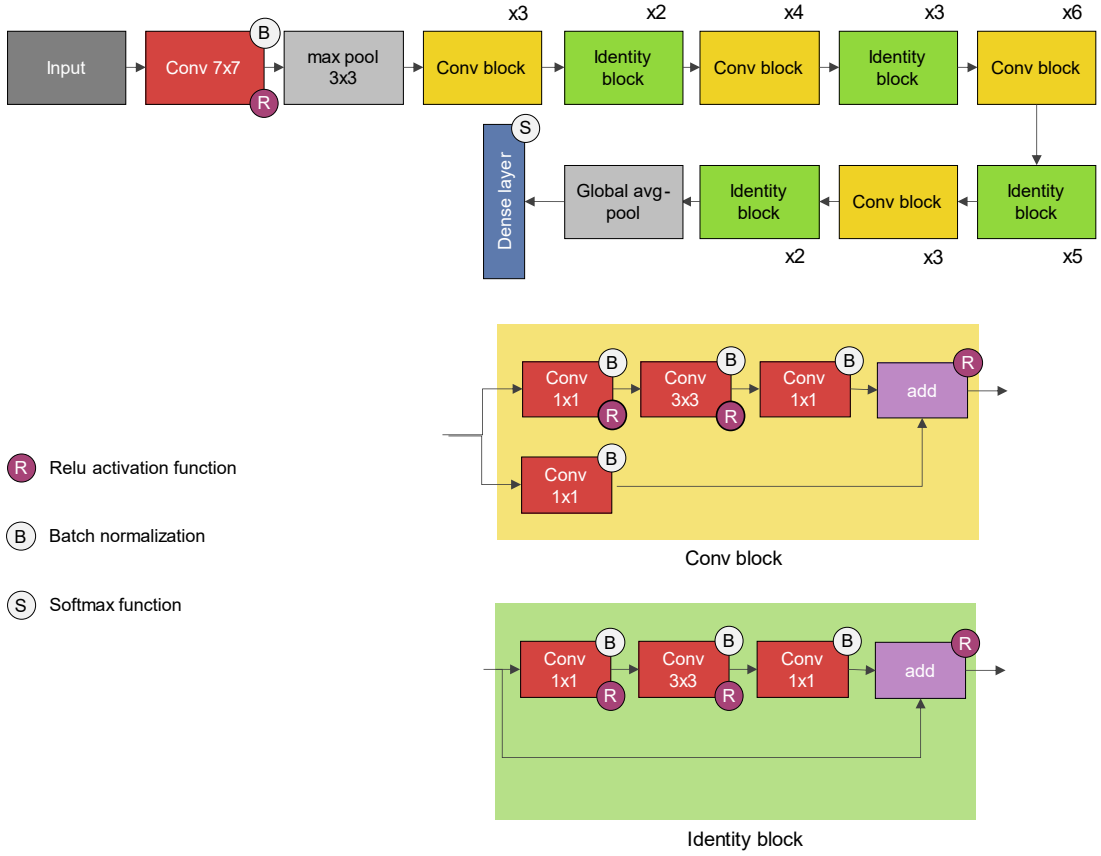


Fig. 10 – ResNet-50. [Based on: He et al., 2016; Karim, 2019]

network to learn. The name ResNet originates from the technique design to combat this problem - “residual learning”. The idea behind this are shortcut connections, add the output of a layer, to a layer ahead, skipping over others. This identity mapping does not have any parameters, its only purpose is to add this output *x*, which does not always have the same dimensions as *F(x)*, due to convolutions and pooling in the layers between. The solution to this is linear projection: either perform padding with zeros, or match the dimensions with 1x1 convolutions (He et al., 2016; Oommen, 2020).

The specific model for the present work is ResNet50-FPN (Fig. 10), since it has good trade-off between accuracy, training time and memory when compared to other backbones supplied by Pytorch (Paszke et al., 2019). The number 50 refers to the number of layers in the network – During the authors research several Resnet architectures in which models with fewer layers were less accurate, and deeper models (He et al., 2016).

To avoid imbalanced weights – with extremely high or low values – which can influence the training process, a normalization method is frequently implemented in ANNs. Instead of normalizing the input of the network, batch normalization can be applied to a layer, in which the output from activation functions is normalized. This method has the advantage of also implementing variables that are “learnable” and updated per training batch. Additionally, this technique increases training speed (Ioffe and Szegedy, 2015).

2.2.2.2. Neck

Object detection for different size objects can be challenging. Object detectors like the FRCNN used multiple anchor boxes on a feature map to detect them which is faster, but less accurate, than other alternatives used at a time - A pyramid of features would be more accurate than this process, but featurizing each level of an image would

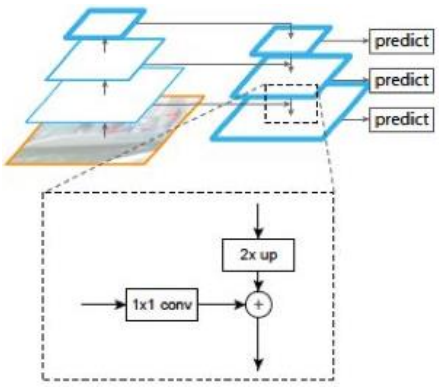


Fig. 11 – FPN with lateral connection. [Source: Bochkovskiy et al., 2020]

increase inference time considerably. Feature pyramid networks (FPNs) were proposed by Lin et al. (2017). FPN is composed of a bottom-up pathway, a top-down pathway and lateral connections, which allow the combination of semantically strong, low-resolution feature, with semantically weak, high-resolution features. The bottom-up pathway is simply a feature extractor, like the ResNet50. Fig. 11 displays the FPN and the lateral connection. As portrayed, the top-down pathway upsamples the spatial resolution by a factor of 2, and merges it with the corresponding bottom-up map – after a 1×1 convolutional layer to reduce channel dimensions – by element-wise addition (Lin et al., 2017).

2.2.2.3. Head

The detector of Faster R-CNN is the Fast R-CNN, which includes an FCL. Since these types of layers only take fixed inputs, proposed regions should be transformed to match it. RoI pooling takes de proposed regions from the RPN and takes portions of the feature map from the neck corresponding to each region and converts that section into the fixed dimension.

The softmax function is commonly used in ANN, since it allows to compute each class probability from a vector of numbers (Raschka and Mirjalili, 2019). In Faster R-CNN softmax is used both for the RPN and the detector (Fig. 10 and Fig. 12) (Ren et al., 2016). Another function found in almost any CNN is the Rectified Linear Unit activation function, ReLu for short, which tries to solve the vanishing gradient problem by considering 0 any input equal or lower to 0 but keeping its value if its greater than that (Raschka and Mirjalili, 2019).

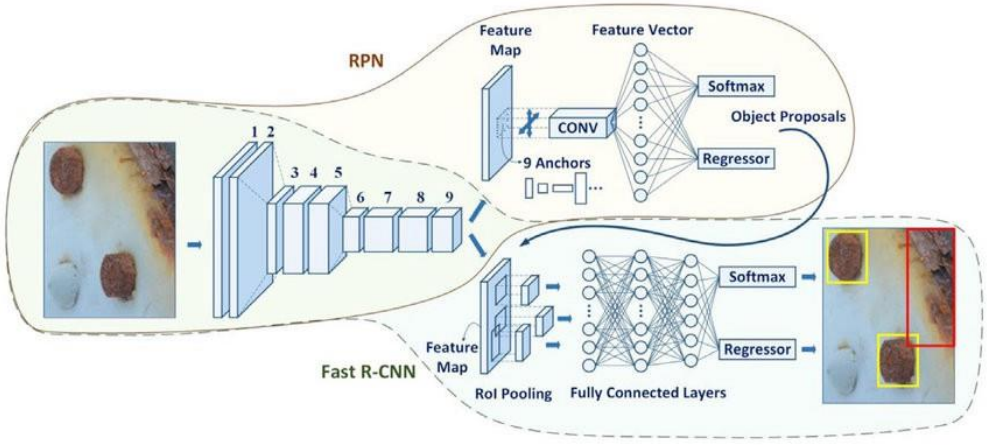


Fig. 12 – Original Faster R-CNN architecture [Source: Khazri, 2019].

2.2.3. YOLO

The previous algorithms discussed were two-stage detectors, since they first extracted regions to locate the objects, instead of using the complete image. You Only Look Once (YOLO) is an object detection algorithm which the first version was released in 2016 by Joseph Redmon, also known as pjreddie, Santosh Divvala, Ali Farhadi and, once again, Ross Girshick. It was the first of its kind, a single neural network that predicted bounding boxes and class probabilities directly from full images in one evaluation, a one-stage detector. In summary, an input image would be divided into an *SxS* grid and for each grid cell, *B* bounding boxes are predicted as well as a confidence score for each one (Fig. 13). These scores reflect the confidence of the model that this bounding box contains the object. Additionally, for each cell, the network would predict the class of the object. At the time, YOLO was extremely fast, it could process streaming video in real-time and, compared to Fast R-CNN, it would wrongly classify the background as an object only half of the times. However, although it would quickly identify objects, it would struggle with accuracy, particularly in small objects (Redmon et al., 2016).

Redmon would later work alongside Farhadi to improve this network and release YOLOv2, or YOLO9000, in 2016 and YOLOv3 in 2018 (Redmon and Farhadi, 2018, 2016). In February 2020, Redmon announced in his personal twitter account that he would abandon CV research due to concerns regarding the usage of his work for military applications and privacy violation (Joseph Redmon, 2020). However, his work would be picked up by Bochkovskiy et al. (2020) and in April 2020, YOLOv4 would be released. Only a few months later, in June 2020, Ultralytics released a new version of YOLO, which they designated as YOLOv5 (Jocher et al., 2021). However, this announcement has caused immense controversy in the machine learning community platforms particularly because the official YOLOv5 paper is yet to be released as of September 2021, resulting in reservations regarding the naming of the model, since the apparent main differences from YOLOv4 are its implementation in PyTorch and data augmentation techniques, and regarding the legitimacy of accuracy and reproducibility of the announced results (Nelson et al., 2020; Supeshala, 2020). However, multiple studies have been made using this network – Detecting apples in orchards (Kuznetsova et al., 2020), mask wearing recognition (Liu, 2020), detecting heavy goods vehicles (Kasper-Eulaers et al., 2021), and more. Due to the lack of consensus regarding this legitimacy of this version of YOLO, YOLOv4 was implemented during this work.

YOLOv4 brings the concepts of Bag of Freebies (BoF) and Bag of Specials (BoS). In the BoF are methods that improve the model's performance, without introducing extra computational cost during inference (Zhang et al., 2019). Modules and methods that can improve accuracy but slightly increase inference cost are included in the BoS (Bochkovskiy et al., 2020).

2.2.3.1. Backbone

The backbone, or feature extractor, of YOLOv4 is composed of the CSPDarknet53, and includes a BoF and BoS.

In YOLOv4 the BoF consists of CutMix and Mosaic data augmentation, DropBlock regularization and Class label smoothing (Bochkovskiy et al., 2020). Data augmentation was developed to reduce overfitting, through several transformations it produces new data from the original training data (Shorten and Khoshgoftaar, 2019). In CutMix, patches from different training images are cut and pasted, where the ground truth labels are also mixed proportionally to the area of the patches (Yun et al., 2019). Mosaic simply puts together 4 different images into a synthetic new one (Hao and Zhili, 2020). Another way to reduce overfitting, would be to produce images which have some parts missing. This is known as dropout. Instead of dropping out features randomly, since in images features are correlated spatially, DropBlock drops features in adjacent regions of a feature map. As it discards features in a correlated area, the networks must consider the available parts for evidence to fit the data (Ghiasi et al., 2018).

Often detection networks use a probability distribution over the classes using the softmax function, for each object. During training, this value is compared to the ground-

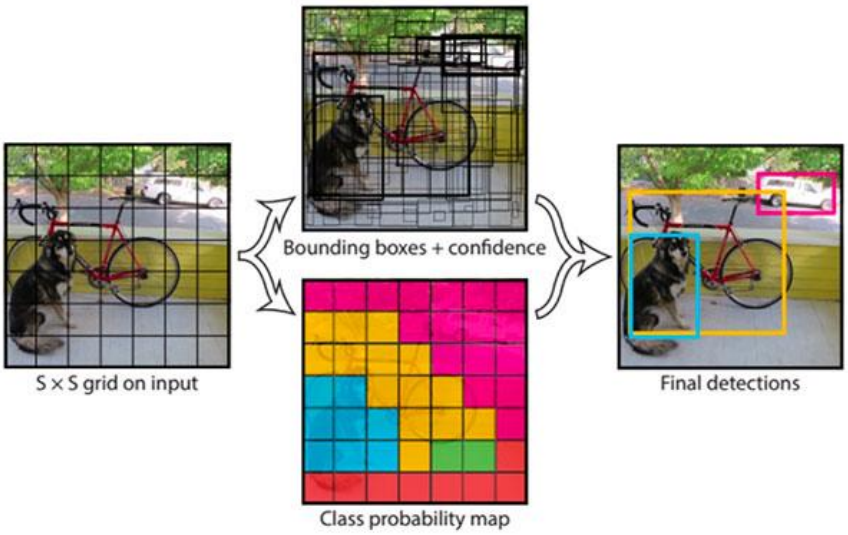


Fig. 13 - YOLO detection [Source: Redmon et al., 2016]

truth, which is usually one-hot encoded – the true class has a probability of 1 and the other have 0. This can cause the model to be overconfident (Zhang et al., 2019). Szegedy et al. (2015) proposed class label smoothing to avoid this problem.

As for the BoS in the backbone, the methods are Mish activation, Cross-stage partial connections (CSP), and Multi-input weighted residual connections (MiWRC) (Bochkovskiy et al., 2020).

CSPDarknet53, the backbone of YOLOv4, is a CNN that uses DarkNet-53 and employs a CSP strategy to partition the feature map (Wang et al., 2019) Darknet was part of YOLOv3 and its name derives from the 53 convolutional layers (Redmon and Farhadi, 2018). Its architecture is presented in Fig. 14. CSPDarknet53 is based on DenseNet, developed by Facebook AI Research, and characterized by its dense layers that reuse features by concatenating the output of a layer with is convolution (Fig. 15a). The motivation to use these layers is to avoid the vanishing gradient by reducing the number of network parameters, which increases along with the depth of the network (Huang et al., 2018; Jacob Solawetz, 2020). CSP is a follow-up to this concept. Instead of using the full output of a layer and pass it on to the next one, the output of a layer will be copied and divided into two parts: one will pass through a dense layer, and the other will go to a partial transitional layer and be concatenated with the output of a dense block. In a dense layer, the input is a concatenation of the part of the output of the previous

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Table 1. **Darknet-53.**

Fig. 14 - Darknet-53 architecture [Source: Redmon and Farhadi, 2018].

layer, as mentioned, and the result of a convolutional. Fig. 15b showcases this method employed to a DenseNet (Wang et al., 2019).

The concept of multi-input weighted residual connections (MiWRC) was introduced with EfficientDet, a CNN developed by a Google research team. The main goal is to fuse features with different resolutions. However, the authors noted that, since different input features are at different resolutions, they usually contribute to the output feature unequally. Hence, they proposed assigning them weights, which can be trained and, consequently, optimized by the network (Tan et al., 2020).

2.2.3.2. Neck

The neck are layers between the backbone and the head that collect feature maps from the backbone at different levels. In YOLOv4, after the backbone is added a Spatial Pyramid Pooling (SPP) module, and Path Aggregation Network (PAN) as a method of parameter aggregation between the backbone and the head (Bochkovskiy et al., 2020).

Early on, most CNNs required a fixed input image size and, when the input does not meet this ratio, it can be cropped or warped. However, this might result in the loss of part of the objects or severe distortions, compromising accuracy. The reason why CNNs require a fixed input is not due to the first part of network, the convolution layers, but rather the FCLs that follow them. A SPP module are layers added after the last convolutional layer, in which the feature maps are spatially divided into bins. Then a

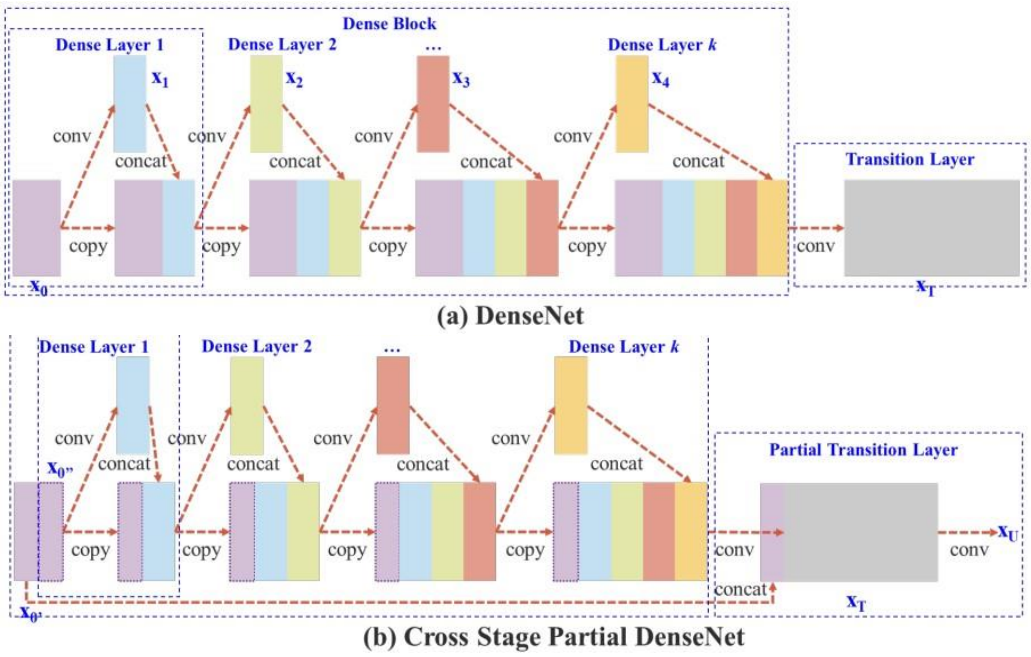


Fig. 15 – DensetNet and CSPDenseNet architectures [Source: Huang and Wang, 2019].

maximum pool is applied to each bin for each channel, originating a 1-D vector. (He et al., 2014). In YOLO, however, the SPP is modified to firstly apply max-pooling of varying kernels, instead of creating bins, and afterwards concatenate these feature maps (Huang and Wang, 2019).

PAN is an evolution of the FPN used in FRCNN, which suggests a bottom-up path in addition to the top down path used in FPN and tries to propagate low level information to the top by also creating skip connections between low level feature maps and top ones (Liu et al., 2018). Instead of the original method which adds these layers, in YOLOv4 they are concatenated (Bochkovskiy et al., 2020).

2.2.3.3. Head

The head, also known as the detector, makes a final prediction – a vector with the coordinates of the predicted bounding box, the confidence score of the prediction and the label. YOLO applies the head network to each anchor box. In YOLOv4, the head is its predecessor YOLOv3, which implements the method summarized previously of dividing the input in a grid. Here, the BoF is made of CIoU-loss, Cross mini–Batch Normalization (CmBN), DropBlock regularization, mosaic data augmentation, Self-Adversarial Training (SAT), elimination of grid sensitivity, multiple anchors for a single ground truth, cosine annealing scheduler, genetic algorithms for selection of optimal hyperparameters and random training shapes (Bochkovskiy et al., 2020).

From the features map created by the convolution layers, several anchor boxes of different ratios are created to represent of objects of varying sizes. The IoU-loss (Intersection over Union), is the formula for the difference between the predicted and the real bounding box (Fig. 16) (Bochkovskiy et al., 2020). The Complete IoU (CIoU), besides the overlapping area, also considers the central point distance and the aspect ratio (Zheng et al., 2019).

Since batch normalization does not perform well when the batch size is small, because it does not reflect the similar statistics of the full training set, cross iteration batch normalization was developed, which considers $k-1$ most recent iterations (Yao et al., 2020). CmBN is a modified version of this method that considers statistics between mini-batches (Bochkovskiy et al., 2020).

SAT is a data augmentation technique that operates in 2 forward-backward stages. On the first pass through the network, instead of altering its weights, the network alters the original image by adding noise to it, which is described as an adversarial attack of the network on itself. Afterwards, the network is trained to detect an object on this

modified image as it normally would, with the original label. This can make the model more robust to noise (Bochkovskiy et al., 2020).

Grid sensitivity is a problem introduced in previous versions of YOLO but solved in YOLOv4 where the model struggled to predict bounding box centers that are in a grid boundary. This was solved by adjusting the formula to calculate them (Bochkovskiy et al., 2020; Long et al., 2020).

In DL the learning rate is a hyperparameter that determines how much the weights are adjusted in response to the estimated error each time they are updated – If it is too low, the training process can become too long, on the other hand, if this value is too high, it might converge to a sub-optimal set of weights (Kim, 2017). Instead of constantly reducing the learning rate by a specific rate like the FRCNN, the authors of YOLOv4 used a cosine function. It starts by reducing the rate slowly, then very quickly, and finally ends up with tiny slope, reducing small learning rate until it reaches 0 (Loshchilov and Hutter, 2017; Zhang et al., 2019). For other hyperparameters, to try to find their optimal value, genetic algorithms are used. Genetic algorithms belong to the family of evolutionary algorithms, which are algorithms inspired by biological evolution and can be applied to a variety of challenges. Genetic algorithms, however, are often used as optimizers. They select the best models based on the "survival of the fittest" – each phenotype (model) has a set of chromosomes (parameters). In each generation (iteration), the quality of each individual is evaluated, and better phenotypes (models) form a new generation, in each they recombine their genetic information and pass it to their descendants (Whitley, 1994).

Object detectors are frequently trained with a fixed input image shape. To improve generalization, since YOLOv2 multi-scale training has been implemented. Instead of adjusting the images to fit a resolution, every *N* mini-batches, the network

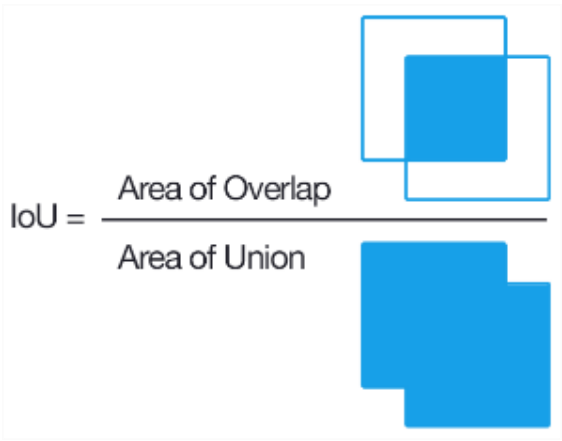


Fig. 16 – Depiction of IoU calculation [Source: Rosebrock, 2016].

randomly chooses a new image dimension size. This makes it more versatile and robust, forcing the network to predict across a multitude of input dimensions (Redmon and Farhadi, 2016).

For the detector, several BoS techniques applied have been utilized in previously such as the mish activation, SPP and PAN blocks. However, the head also implements Spatial Attention Module (SAM) block and DIoU-NMS.

An attention module is, as the name implies, a mechanism parallel to the human concept of attention - focus on important features and suppress the rest. Given a feature map, SAM performs max-pooling and average pooling. Afterwards convolutes these outputs and passes the result trough a sigmoid function that normalizes these values (Woo et al., 2018). Bochkovskiy et al. (2020), however, modified SAM to be point-wise, instead of spatial-wise, by convoluting the feature map, instead of pooling (Fig. 17).

DIoU is an improvement from IoU that also considers the distance between the center points of the bounding boxes. NMS stands for non-maximum suppression and it is used to filter redundant bounding boxes for the same object (Bochkovskiy et al., 2020).

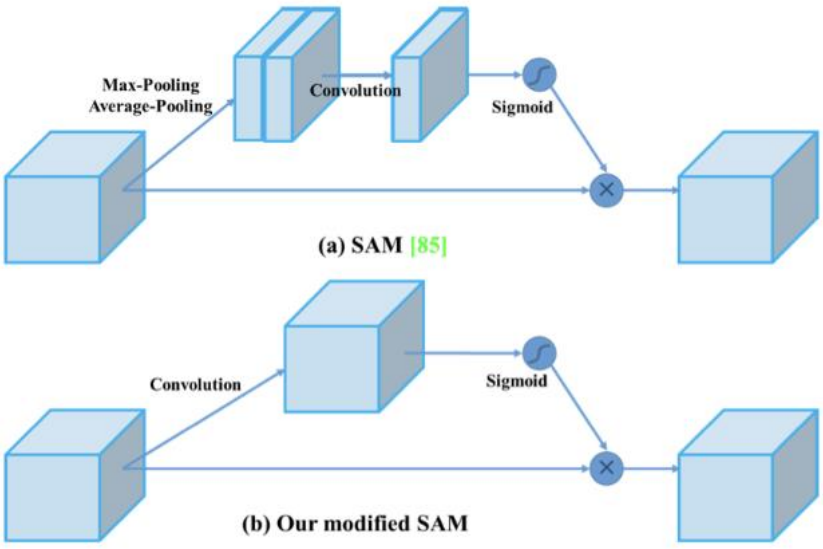


Fig. 17 – a) Original SAM. b) Modified SAM in YOLOv4 [Source: Bochkovskiy et al. (2020)].

2.2.4. Transfer Learning and fine-tuning

So far, several parameters and techniques were presented as well as the hardships when trying to optimize a model. Since training a DL model demands an enormous amount of data, publicly available, high-quality, labeled datasets with images are critical for adapting DL to entomological problems. There have been initiatives like the IP102 (Wu et al. 2019), the IPM image project (Center for Invasive Species and Ecosystem Health, 2018) and the iNaturalist (Van Horn et al. 2018) datasets that have contributed to solving this problem, with the latter not being exclusively for insects. However, some are not easily available and might not have the amount of data necessary. Additionally, none of these are exclusively of sticky traps.

It is reasonable to assume that a network trained with a large dataset, could possibly be applied to other tasks. This concept is termed transfer learning, an incredibly valuable technique, being described by some author as the next driver of machine learning commercial success (Andrew Ng, 2016; Kim, 2017; Ribani and Marengoni, 2019; Shin et al., 2016). In fact, CNN such as Faster R-CNN and YOLOv4 provide already pre-trained versions of the models in different datasets (Bochkovskiy et al., 2020; Paszke et al., 2019). Just in entomology, several studies have tested this approach - Nieuwenhuizen et al. (2018) used a Faster R-CNN pre-trained with MS-COCO dataset to identify tomato pests, also in yellow sticky traps. Zhong et al. (2018) used a pre-trained YOLO in Imagenet Dataset to detect flying insects in yellow sticky traps.

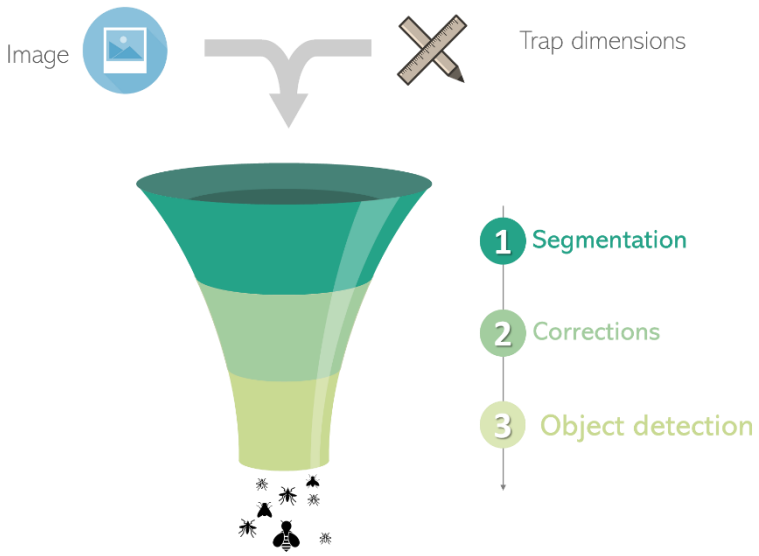


Fig. 18 - Diagram of proposed work.

3. Methodology

3.1. Data acquisition

The data available consists of 313 images from 58 different chromotropic traps, kindly supplied by DRAPN (Direção Regional de Agricultura e Pescas do Norte). The traps originated from different areas across the north of Portugal, particularly from Entre-Douro e Minho and Trás-os-Montes e Alto Douro provinces. Some were collected in person directly from the field, while others were collected from DRAPN facilities.

When the presence of any of the target insects was verified, its location was delimited in different ways: surrounding the object with a marker (Fig. 19a), or drawing a symbol next to it (Fig. 19b), or, to avoid signaling the targets, a photograph of the trap was taken before and after the demarcation. In this case, the marked version was kept separately and used for reference only when annotating bounding boxes (Fig. 19d and Fig. 19e). Occasionally, TE was the only insect present in a trap, as demonstrated in Fig. 20. Here, due to the high frequency of insects on a single trap, with areas where multiple insects where overlapping, or with body parts missing, the registered count was 100, for simplification. Table 1 presents the number of TE and ST found, and number of images containing each one.

The images were obtained using a smartphone with a 12-megapixel camera, and in different lighting, backgrounds and sometimes distorted. To assess how some properties of an image could impact different stages of the process, the trap's real dimensions were recorded and there was binary classification for the presence of certain characteristics (Fig. 21). Each image was also classified as good, reasonable, or bad quality, according to their number and severity (Fig. 22). Table 2 describes the total number of images per characteristic and per quality class. This data was annotated in a .xlsx file with following structure:

- Image number – For example: in IMG_1771.jpg, the number is 1771.
- Width – Real width of the trap, in cm.
- Height – Real Height of the trap, in cm.
- Count – Number of target insects in the image (0 if none). In cases like Fig. 20, the total count of target insects was simply marked as 100.
- Label – Label of the insects counted in the previous variable (TE or ST). Since the traps never had both, this is not an issue.
- Quality – Overall quality of the image (Good, reasonable, or bad).
- Reflections – Presence of specular reflection (1 - true, 0 - false).

- Shadows – Presence of any shadow (1/0).
- Cut_margins – If the image does not contain the full trap, but rather cuts off some of the margins or corners (1/0).
- Distortions – Presence of perspective or other distortions (1/0).
- Poor_lighting – If the overall quality of the lighting in the image is not good (1/0).
- Blured_noise – Presence of noise or if the image is blurred (1/0).
- Yellow_toned_background – If the background behind trap is yellow-toned or if there are any yellow objects (1/0).
- Transparency – If the trap itself is somewhat transparent (1/0).
- Plastic – If the trap is enveloped in plastic (1/0).
- Segmentation – If the segmented process was successful (1/0).
- Correction – If the correction of distortions was successful (1/0).

Table 1 - Number of images and number of insects per species.

	Number of images	**Total count of insects**
Trioza erytreae	55	1661
Scaphoideus titanus	29	173

a) ***b)***

c) ***d)***

Fig. 19 - Various forms of annotation. a) ST circled with a red marker. b) TE marked with red dots next to it. c) trap with ST but not marked. d) Annotated version of e), kept in a folder and used as reference for bounding boxes.

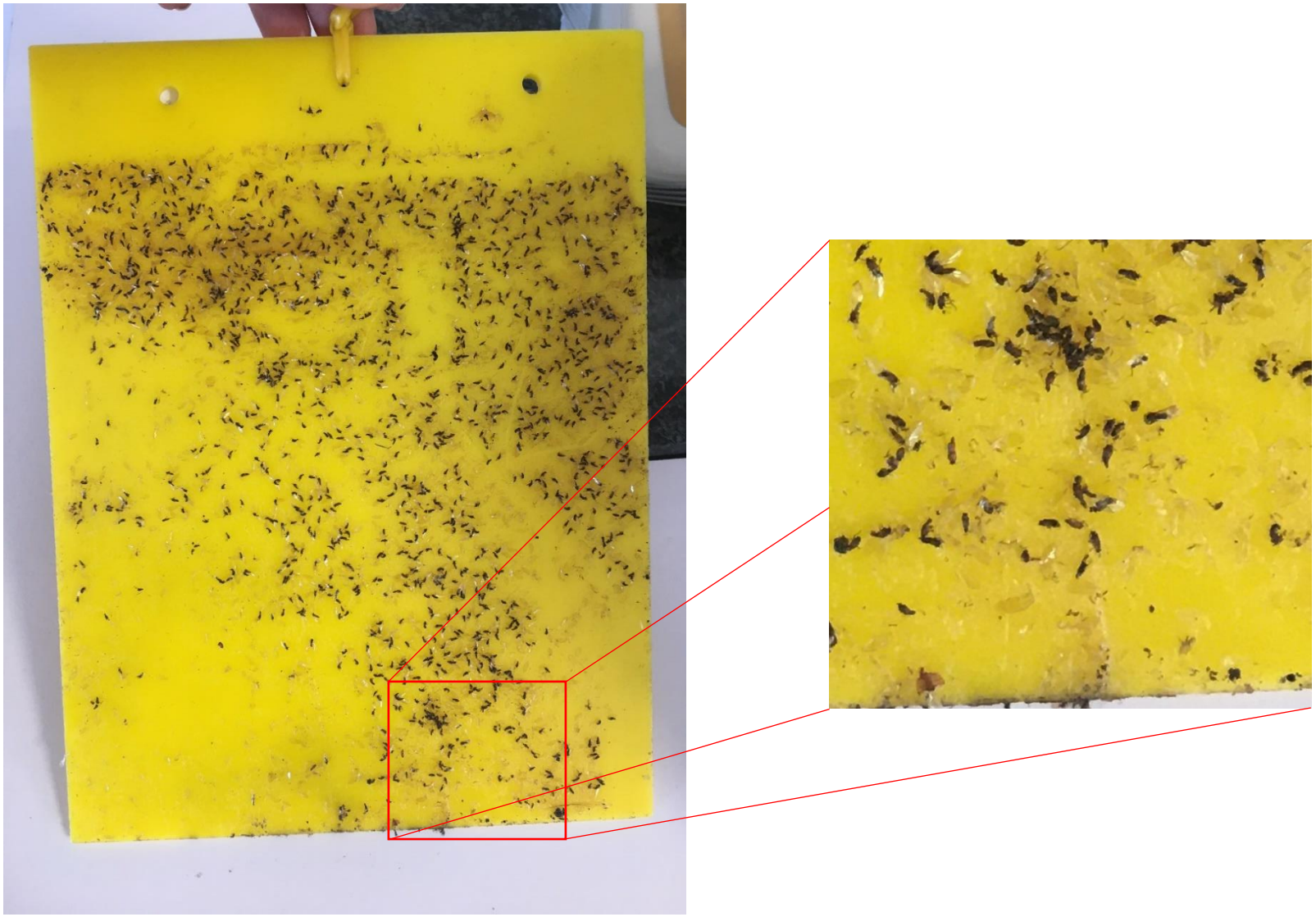


Fig. 20 - Chromotropic trap with high numbers of TE.

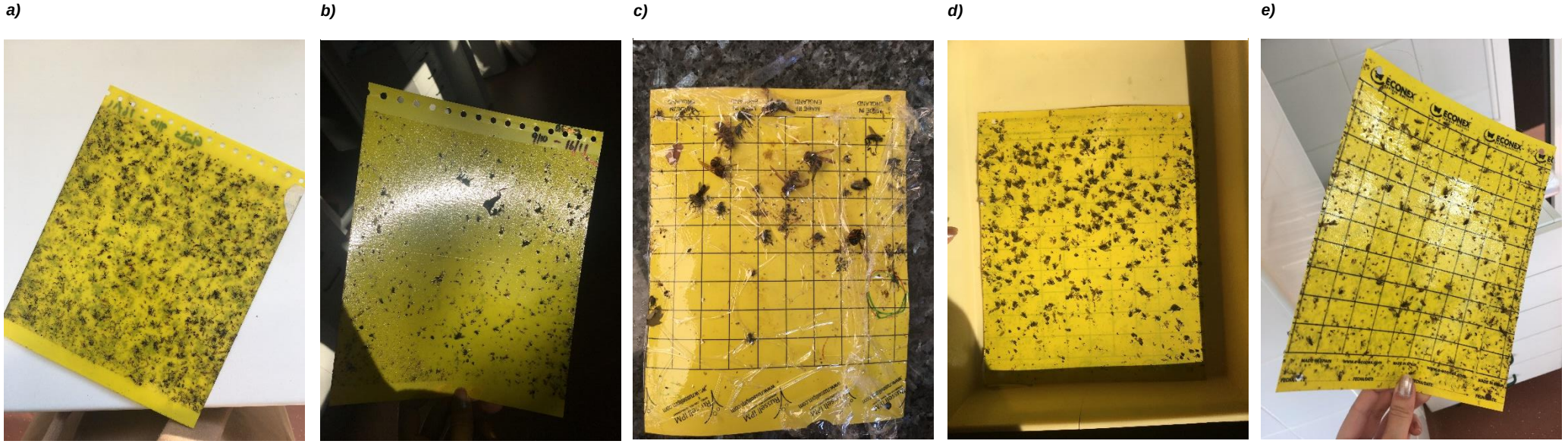


Fig. 21 – Examples of different images properties. Image a) has cut margins and transparency, b) has shadows, specular reflections, overall poor lighting and is distorted. Images c) and d) are only positive for plastic and yellow-toned background, respectively. Image e) is distorted and has specular reflection..

Fig. 22 - Images classified has good (left), reasonable (center) and bad (right).

3.2. Segmentation

Segmentation is the process of partitioning an image into regions, i.e. sets of pixels, that share similar characteristics such as color, texture and intensity (Gonzalez and Woods, 2008). It is considered one of the most crucial steps in image processing since failed segmentation almost always implies a failure in the further steps of the process. In this project, the segmentation goal is to obtain the mask of the full yellow trap, removing all the background. Segmentation was employed using MATLAB (version R2020b) (MATLAB, 2020).

Since the traps have a very strong yellow color, a color-based segmentation was implemented. Some techniques to pre-process the image to enhance and adjust the color and lighting were tested. However, since they did not achieve good results, these were discarded.

MATLAB's Color Thresholder was used since it allows testing of several thresholds for different color spaces (Fig. 24). The developed function *segmentation.m* takes as input the original image, and paths to save its mask and the corresponding binary image. The full segmentation process is visually described in Fig. 25.

The segmentation process consisted of converting each image to two different color spaces: HSV and YCbCr. HSV correspond to Hue, Saturation and Value. Hue describes a color in a 360º spectrum, saturation is intensity from grayscale and value corresponds to brightness. YCbCr color space can separate luminance, which is light intensity, from chrominance, a light wave with color Cyan Red and Cyan Blue, effectively (Zaidi et al., 2015).

The HSV image was segmented by thresholding the Hue channel, considering pixels between 0.110 to 0.230, which correspond to the yellow hue. The YCbCr segmentation was implemented by thresholding the second channel, with a minimum

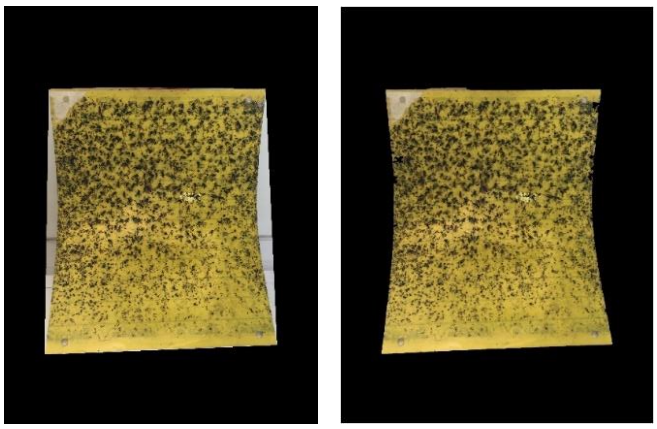


Fig. 23 - Convexhull (left) compared to boundary method (right).

threshold equal to mean value of the pixels on this channel. A binary image resulting of the intersection between the HSV and YCbCr segmentations, was then obtained. Afterwards, a morphological closing is applied, so that other objects are separated and to reduce some noise. Since, ideally, smaller objects are now disconnected from the main one – the trap- all the objects are removed, except from the largest one.

To retrieve not only the yellow region but the full trap and content, the object boundaries are traced, and everything inside is considered true. Contrarily to the convex hull, this method does not trace exclusively straight lines and, therefore, does not include unnecessary background pixels that can negatively impact point detection and consequently, geometric transformations. However, it can sometimes exclude parts of the image that were not well segmented. Finally, the mask is obtained from the binary image.

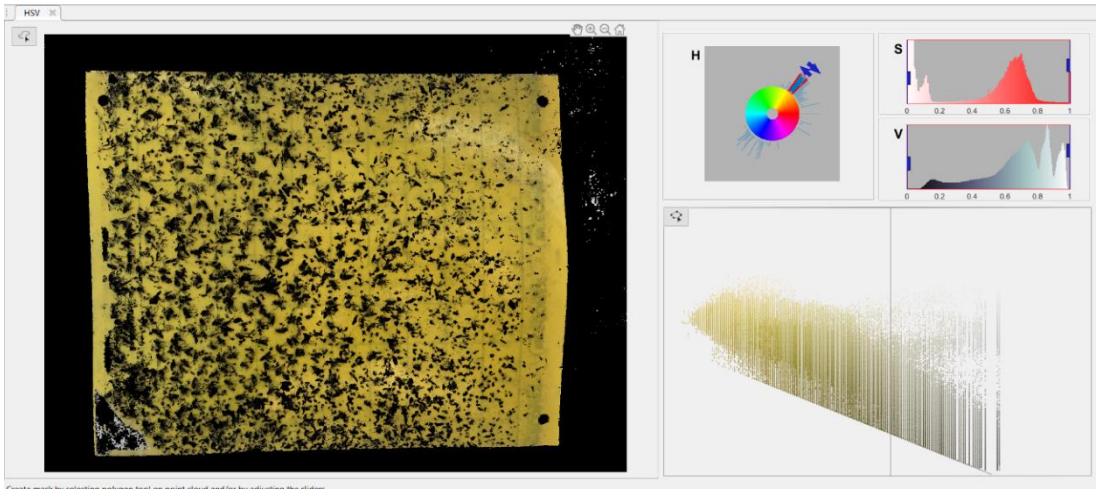


Fig. 24 - Example using MATLAB's Color Thresholder for the HSV color space.

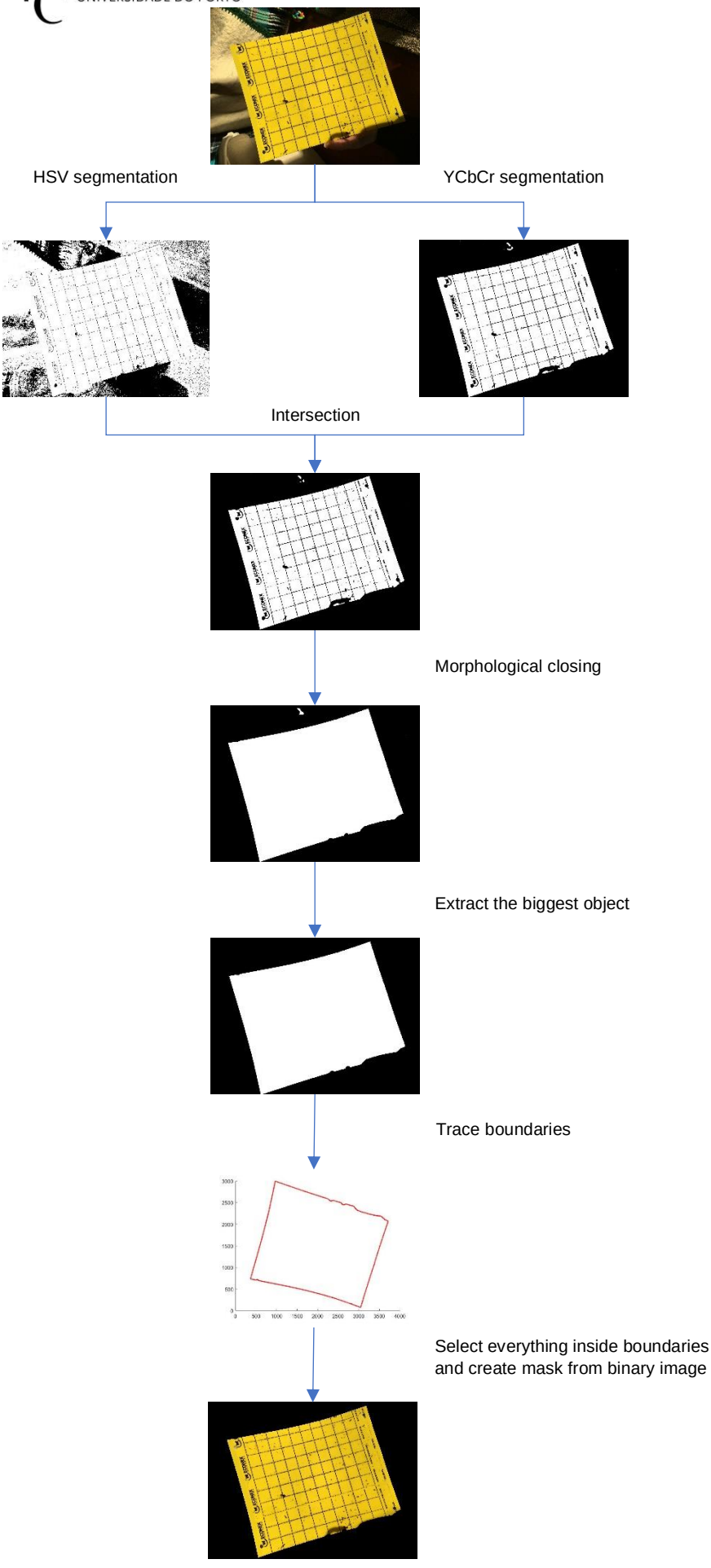


Fig. 25 - Diagram of segmentation process.

3.3. Geometric transformations

Ideally, the traps should be presented in an image as a rectangle, with little to no distortions. However, this is usually not the case. These traps are often distorted from being strongly tied to branches or other surfaces, or simply by the pressure applied while holding them for the camera. Additionally, images are often not taken with the trap perfectly centered and parallel to the camera. These distortions that might compromise the size and ratio of objects, which in consequence could impact detection. Therefore, an algorithm was developed to overcome this (Fig. 27).

After the segmentation, ideally, there should only be one main object in the image, the full trap. Therefore, the following approach does not take into consideration the presence of any other objects.

Firstly, using *regionprops*, a MATLAB function thar returns different properties of image regions, the orientation, major axis length and minor axis length are obtained. The minor and major axis lengths correspond to the length (in pixels) of the minor and major axis of the ellipse that comprises the object. The orientation of the object is the angle between the x-axis and major axis of that same ellipse. This value can range from -90 degrees to 90 degrees. Hence, the angle for which the image should be rotated so that the major axis is perpendicular to the x-axis is simply given by equation (1).

$$angle = 90 - orientation \tag{1}$$

The newly obtained *angle* is then used to rotate the image, using MATLAB's function *imrotate* with default parameters. When rotated by an angle θ, each point (x_1,y_1) will become (x_2,y_2) by following the equation (2) (Gaster, 2013).

$$\begin{aligned} x_2 &= \cos(\theta) * (x_1 - x_0) + \sin(\theta) * (y_1 - y_0) \\ y_2 &= -\sin(\theta) * (x_1 - x_0) + \cos(\theta) * (y_1 - y_0) \end{aligned} \tag{2}$$

To perform any geometric transformation, two types of points should be provided: moving points and fixed points. Fixed points are the target coordinates for which the corresponding moving points should be transformed to. The total number of fixed and moving points is 8- 4 corners and 4 mid points between each pair of corners.

Taking in consideration the trap's real width and height, taken during the data acquisition stage, the values for fixed points are obtained using the previously obtained minor axis length as reference. This will also reduce the image size and, but for the

purpose of this work it is not an issue since both networks take as input 416x416 images. As for the moving points, the four corners are retrieved by firstly using the function bwboundaries, which will trace region boundaries in the image. These pixel coordinates will be converted into polar coordinates, in which each point is determined by the distance from the centroid (center of the object) and an angle. Since the major axis is now perpendicular to the x axis, it is assumed that each of the four corners is one quadrant. Hence, each point corresponding to a corner is obtained by obtaining the coordinates of the furthest point from the centroid for each quadrant (0-90º, 90º-180º, 180º-270º and 270º-360º). The mid points between corners are then localized by tracing a straight line between each pair of corners. However, since the object may present some distortion, this does not always represent the true mid-point, as illustrated in the example in Fig. 26. Therefore, if this is the middle point in a vertical line traced between two corners, the true middle point is found by finding the first and last white pixel in the same line. Subsequently, the middle points between corners that form a horizontal line are found by retrieving the first and last white pixel in the respective column.

Afterwards, a second order polynomial transformation is applied to the image.

The full process is portrayed in Fig. 27.

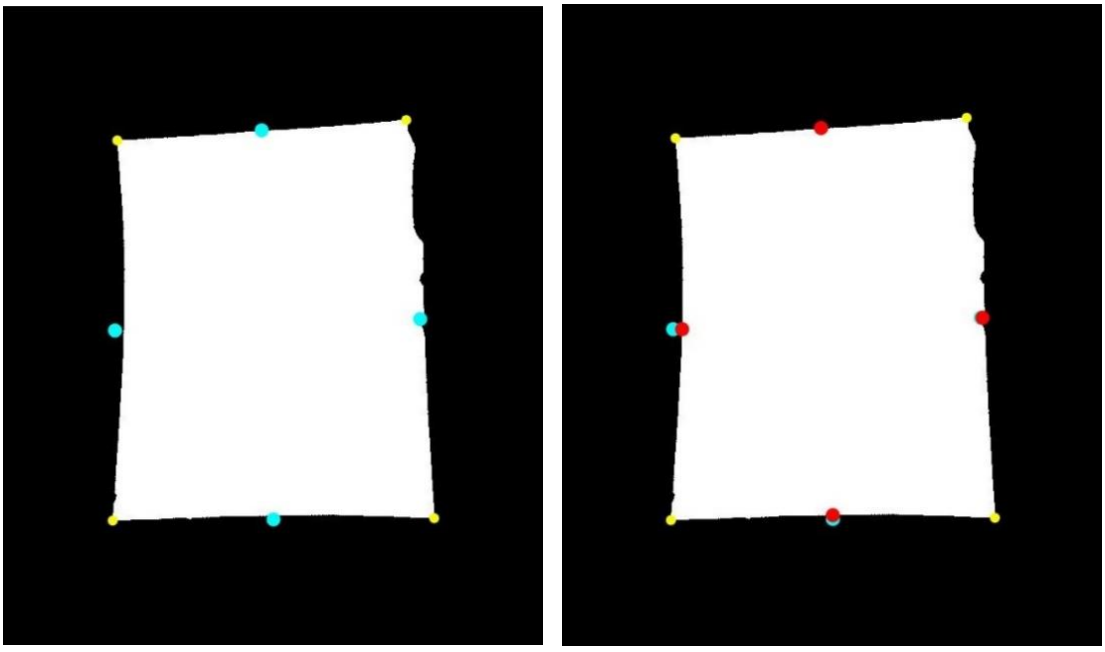


Fig. 26 - Plotted object corners in yellow, the initial estimates for the in cyan blue, and in red are the true mid points.

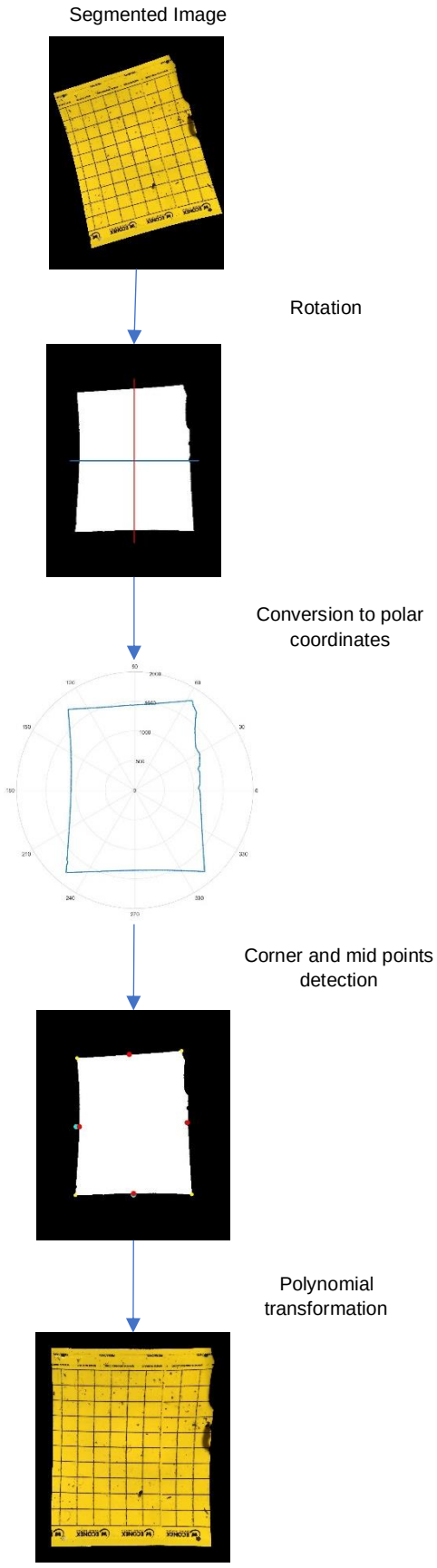


Fig. 27 - Diagram of geometric transformation process.

3.4. Object detection

Both object detection algorithms require different types of annotations, so the software LabelImg (Tzutalin, 2015) was used as an annotation tool, since it allows to save them in different formats (Fig. 28). In images similar to Fig. 20, where there was a large number of TE, only full-bodied and intact insects were considered. Parts of insects or crowded areas were not annotated since whole insects was not perceptible.

For object detection only images of traps with TE or ST were considered. The training set corresponded to 38 images of the data and was the same for both models, and the validation set was of 9 images.

The models were evaluated considering the mean average precision (mAP) on a test set (equal for both models), which is a robust metric for evaluating object detectors that considers precision, recall and IoU.

For a regular classification task, models are usually evaluated considering their precision and recall, which are described below in equation (3) and (4).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (3)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (4)$$

Precision will provide information over the ratio of predicted true positives and the total number of predicted positives, and the recall metric is the ration between true positives

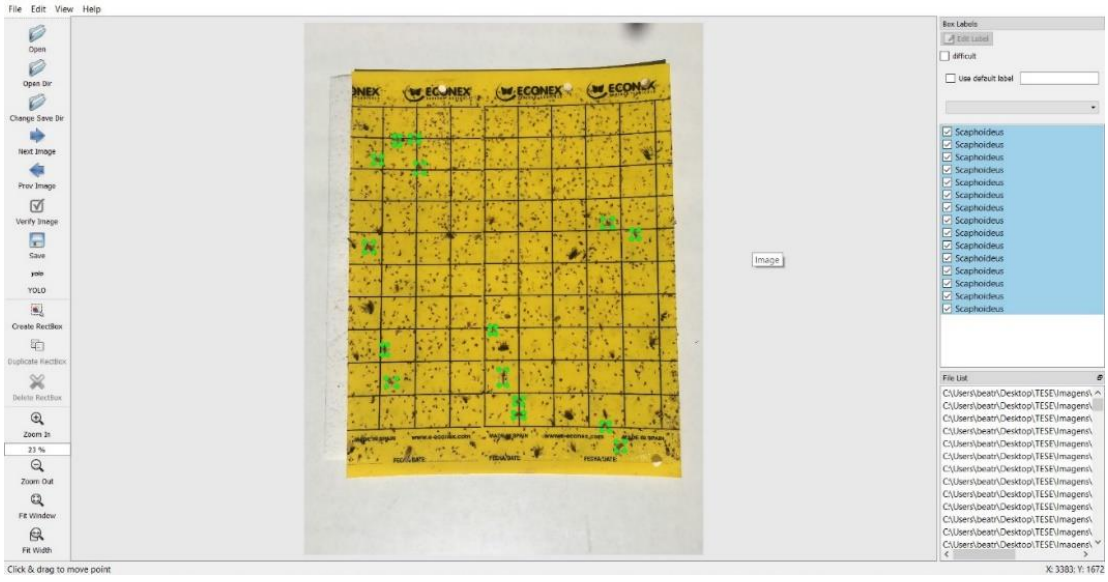


Fig. 28 – LabelImg window.The green points are the corners of the bounding boxes for the objects already labeled.

and ground truth positives (Tan, 2019). In information retrieval, these metrics are respectively described as the fraction of retrieved documents that are relevant to the query, and the fraction of relevant documents that are successful retrieved (Zhang and Zhang, 2009a, 2009b). Precision and recall in document retrieval are described in equation (5) and (6).

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \quad (5)$$

$$Recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \quad (6)$$

In equation (7) the average precision combines recall and precision for ranked retrieval results, for a certain query, the average precision the mean of the predictions scores after each relevant document is retrieved.

$$Average\ Precision = \frac{\sum_r P@r}{R} \quad (7)$$

Where r is the rank of each relevant document, R is total number of relevant documents and P@R is the precision at the top-r retrieved documents (Zhang and Zhang, 2009c).

The task of object detection is parallel to information retrieval. The IoU, as mentioned in the section regarding YOLOv4's BoF, is the overlap between predicted bounding-boxes and the ground truths. A IoU threshold will define what is considered a TP and a FP. This average precision is then calculated for each class, and the mean of those values corresponds to mAP.

3.4.1. Faster R-CNN

The model was implemented using Pytorch, pre-trained trained on ImageNet. Due to computational limitations, batch size was set to 1. With a batch size of 1, the gradient is estimated for each image- known as stochastic gradient descent (SGD) - meaning that although it requires a significantly smaller memory footprint, it is slower and noisier – as it is updated for every instance SGD will keep overshooting, complicating the convergence to the exact minimum (Kim, 2017). However SGD can escape shallow local minima (Buduma and Locascio, 2017; Raschka and Mirjalili, 2019). Additionally, the number of epochs for the model was of 400.

3.4.2. YOLOv4

In YOLOv4 each image should have corresponding text file with the same name that contains for each object the bounding box coordinates and label.

The network installation process was followed according to the instructions available on Alexey Bochkovskiy's github repository (Bochkovskiy, 2020). Contrarily to the previous model, the author advised the number of epochs to be 2000 * number of classes, which in this case corresponds to 4000 epochs. This model was pre-trained on the MS COCO dataset.

4. Results

Fig. 29a displays the total number of images per quality. The percentage of successful segmentations per image quality was assessed and is presented in Fig. 29b, with 80% of images classified as "Good" resulting in a successful segmentation, while the other two classes are barely over half. The properties that contributed the most to segmentation failure are presented in Fig. 30. The percentage of corrected images per segmentation outcome is displayed in Fig. 29c. Here, a successful segmentation results in a correctly transformed image 68% of the times, while a poor segmentation is more likely to result in a bad correction. The great majority of images classified as good, results in a properly transformed image (Fig. 29d). Examples of images, their obtained mask and corrected output, are shown in Fig. 31.

Regarding the detection models, YOLOv4's loss curve of the models trained before and after segmentation and correction, is presented in Fig. 32, while the mean average precision (mAP) for both models is plotted in Table 3. For the YOLOv4 models, the average precision (AP) of both classes is also evaluated.

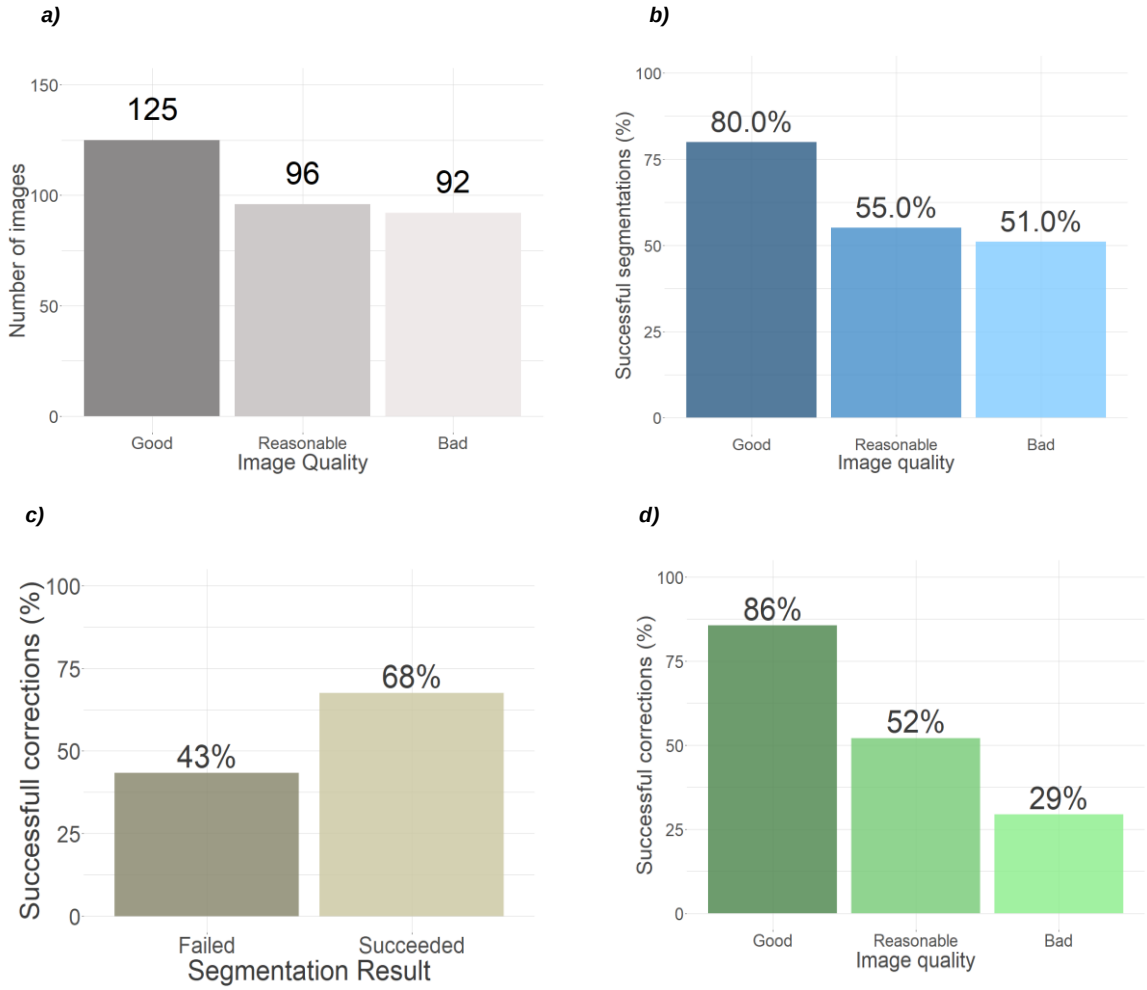


Fig. 29 – a) Number of images per quality class. b) Ratio of successful segmentations per image quality class. c) Percentage of successful geometric corrections by segmentation result. d) Successful geometric corrections per image quality.

Table 2 - Total number of images with each feature and quality.

	Reflections	*Shadows*	*Cut Margins*	*Distortions*	*Bad lighting*	*Blur*	*Transparency*	*Yellow background*	*Plastic*	*Total*
Bad	29	17	24	53	34	4	72	6	5	244
Reasonable	27	7	18	22	6	3	51	17	3	154
Good	10	2	3	6	1	3	36	1	0	62
Total	66	26	45	81	41	10	159	24	8	460

Fig. 30 – a) Ratio of segmentation results per image feature. b) Ratio of geometric correction results per image feature.

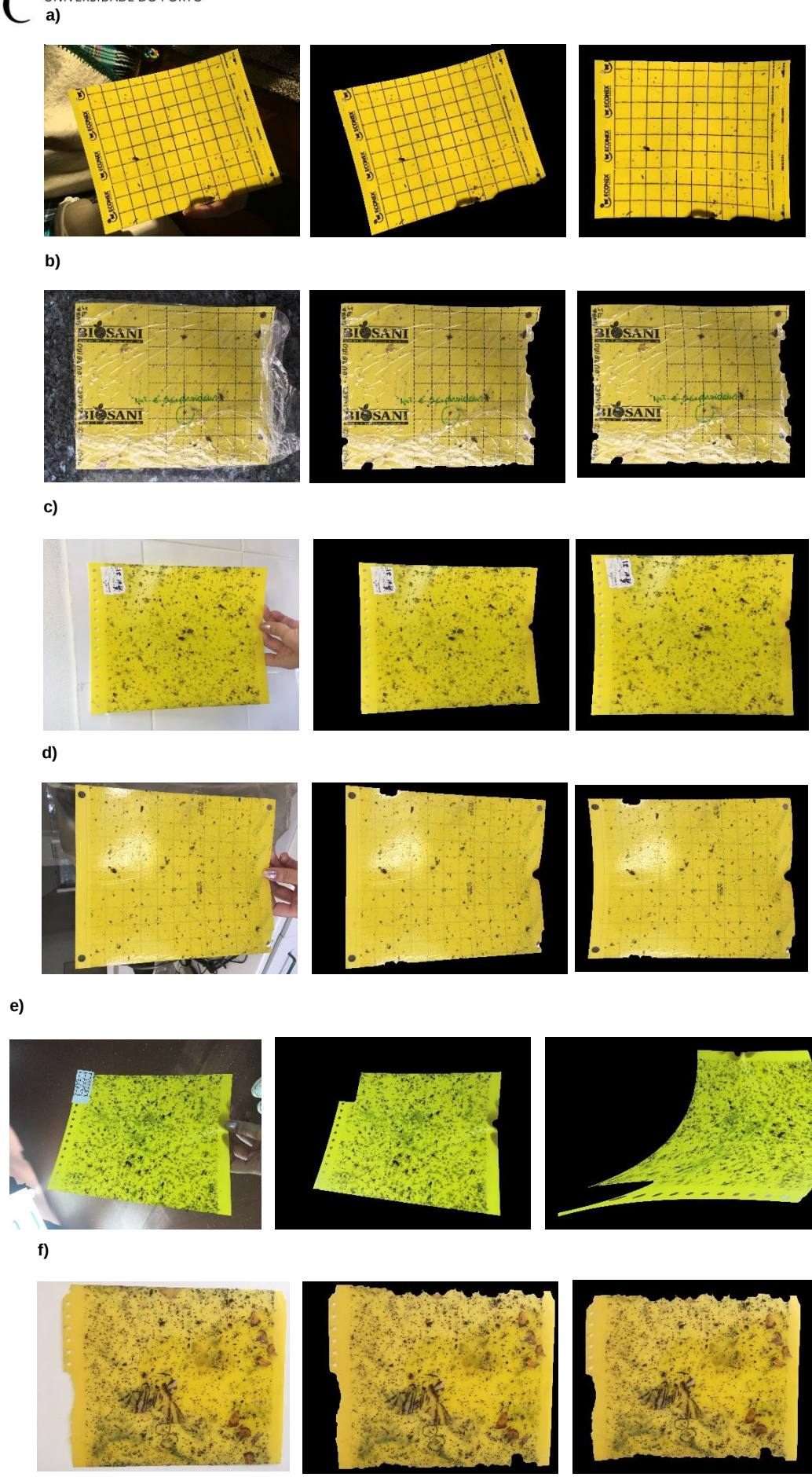


Fig. 31 - Example of original images (left), their mask (center) and after geometric correction (right). a) Image with good results both in segmentation and geometric correction. b) Outputs of a trap involuted in plastic. c) Image of trap with a paper tag that does not touch the trap's margins. d) Trap with intense specular reflection. e) Trap with paper tag that passes its margins. f) Damaged trap.

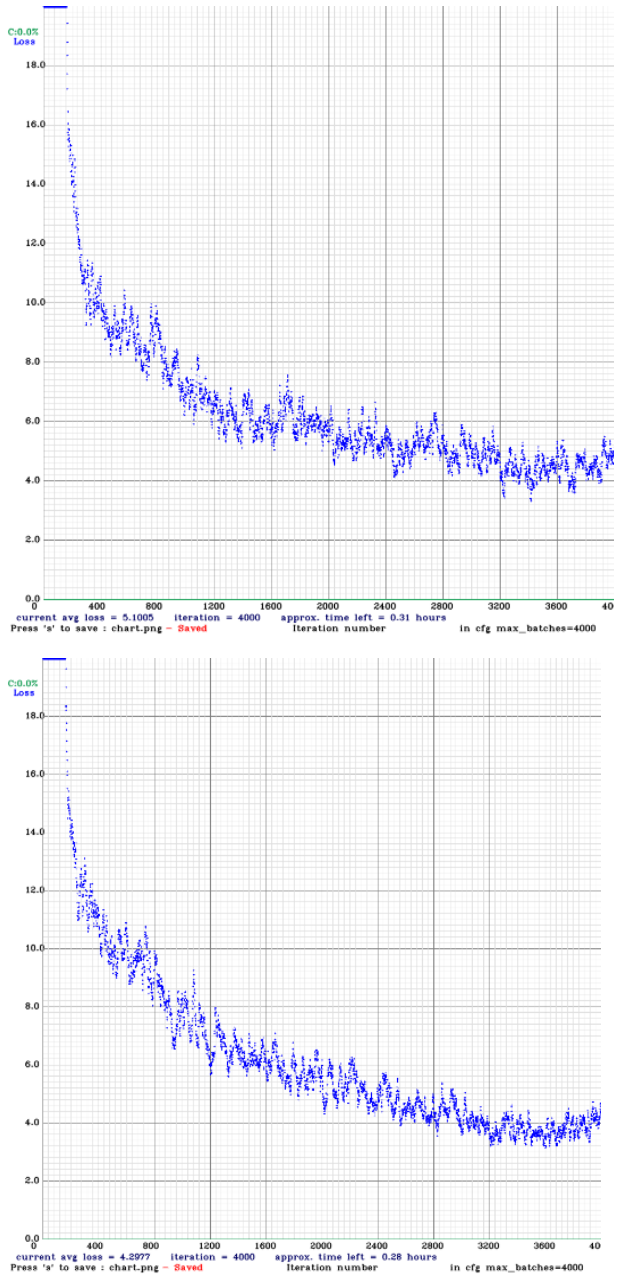


Fig. 32 - Loss curves of YOLOv4 trained in the original images (top) and trained on the processed images (bottom).

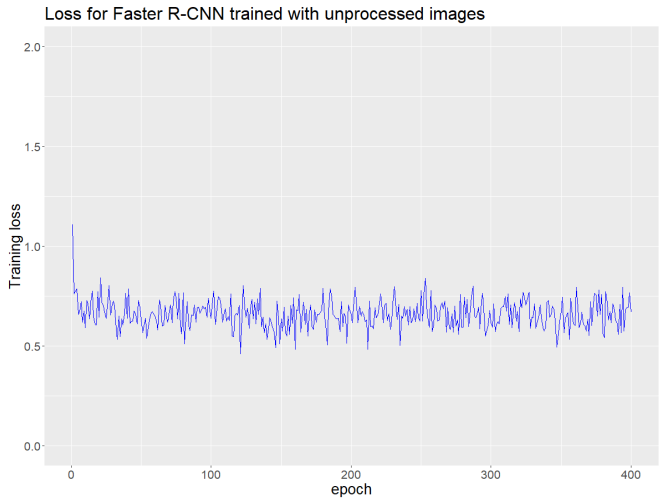


Fig. 33 - Training loss for Faster R-CNN model trained with unprocessed data.

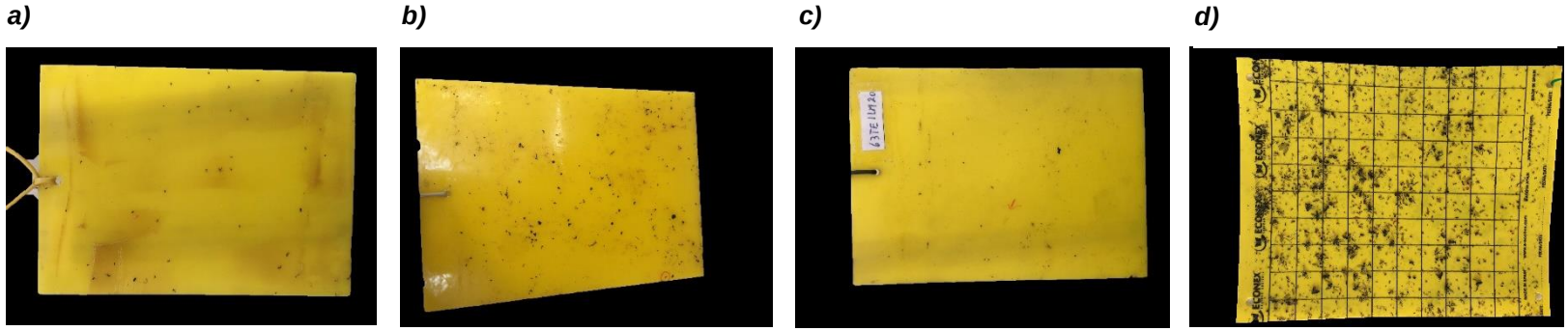


Fig. 34 - Chromotropic traps tied with different colored straps. a) Yellow strap. b) White strap. c) Black strap. d) Green strap

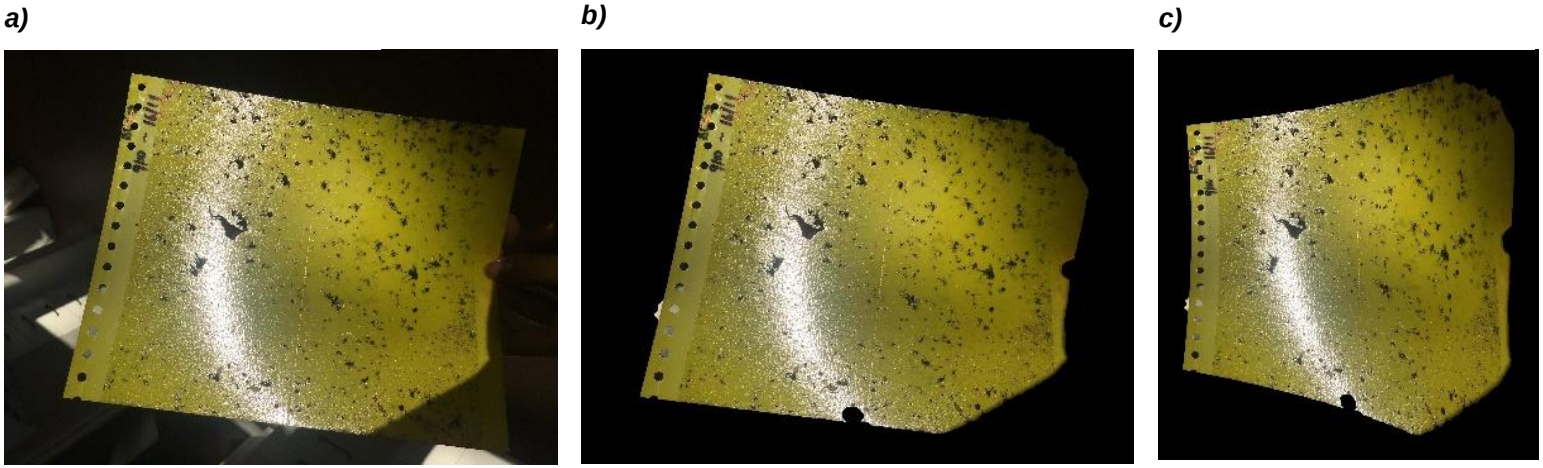


Fig. 35 – a) Original image with poor lighting, severe specular reflection, and shadows. b) Segmentation output. c) Geometric correction output.

Table 3 - Results of object detection models.

Model	Metric	Unprocessed images	Processed images
YOLOv4	AP (TE)	24,79%	25.51%
	AP (ST)	16.65%	4.29%
	mAP@0.5	20.72%	14.90%
Faster R-CNN	mAP@0.5	1,27%	1,46%

5. Discussion

Most of the images collected seem to fall under the “good” or “reasonable” label (Fig. 29a). Not surprisingly, bad quality images seem to have significantly higher number of features compared to them, as shown in Table 2. Higher quality images result more often in successful segmentations (Fig. 29b) and corrections (Fig. 29d). Also, a successful segmentation is more likely to result in a successful geometric correction, as indicated in Fig. 29c.

Certain image properties seem to impact segmentation differently. Fig. 30a displays the ratio of segmentation outcome per feature, and it is observable that lower segmentation success rates belong to the features plastic, shadows, bad lighting, and yellow toned background. By cross-referencing this data with Table 2, it is noted that all the 8 images that were wrapped in plastic, were not well segmented, as shown in Fig. 31b. However, this is a very low number of occurrences, so the segmentation could possibly be effective under this condition in some instances. The presence of shadows also appears to heavily influence the segmentation outcome, since 79% of images with this issue, failed to be segmented properly. A yellow-toned background and bad lighting can also be detrimental to this process, since only around 50% of images in these categories had positive outcomes. As described previously, the feature yellow-toned background also includes yellow-toned objects like the straps used to tie the trap to branches and other surfaces. The data available included traps with yellow, white, black, and green colored straps (Fig. 34). The observed results show that the segmentation fails to properly separate only the yellow strap, which is understandable since the segmentation process is color-based.

The data provided suggests that the segmentation outcome can greatly impact the correction of this distortions, which can be explained by wrongly detecting corners in

poorly segmented images, resulting in images with severe distortions, as displayed in Fig. 31e. The trap in this image had a paper tag on one corner, similarly to several others. Although the boundary selection during the segmentation stage is implemented to try to include regions on the trap that become background because they do not fall between the thresholds, if the object is near a corner, it will not be covered by the boundary, so it will be excluded anyway and jeopardize corner detection. Using the convex hull could overcome this, but it will make the correction of distortions obsolete, as it will not trace curves between points, which will not correctly depict the trap's shape and include background that was previously segmented, as shown in Fig. 23. This is also applicable to insects or points of specular reflection that pass beyond or are very close to traps margins (Fig. 31d). If the problem region is further located from the margins, however, this method is successful (Fig. 31c). Using traps with damaged corners and margins, or images under severe lighting conditions, may also lead to distorted images. Although the output of the image in Fig. 31f is not severely degraded, using damaged traps can be unpredictable since it also compromises corner detection. Fig. 35 displays an image with overall poor lighting, severe presence of shadows and specular reflection, in which the poorly segmented outcome resulted in a strongly distorted image.

Both models had a low mAP, but the FRCNN models were particularly low, achieving a score below 2% on the test sets. When comparing the loss charts, the Faster R-CNN achieves lower training loss than YOLOv4 (Fig. 32 and Fig. 33). This data suggest overfitting, possibly due do the small training data size. Although still low, YOLOv4 had a significant higher mAP, which can be explained by all the BoF and BoS that this network implements, such data augmentation methods. For both YOLOv4 models, the AP for ST was lower than TE, despite the difference in size, which can be attributed to the lower number ST across all images. AP for ST also decreased substantially after processing the images, possibly due to distortions caused by the polynomial transformation – which are more prevalent as the object size increases. The 8 points provided for this transformation are possibly not enough to avoid disturbing straight lines. However, in Fig. 33 it is shown that the model trained with processed data achieves lower training loss, specifically around the 3600th epoch.

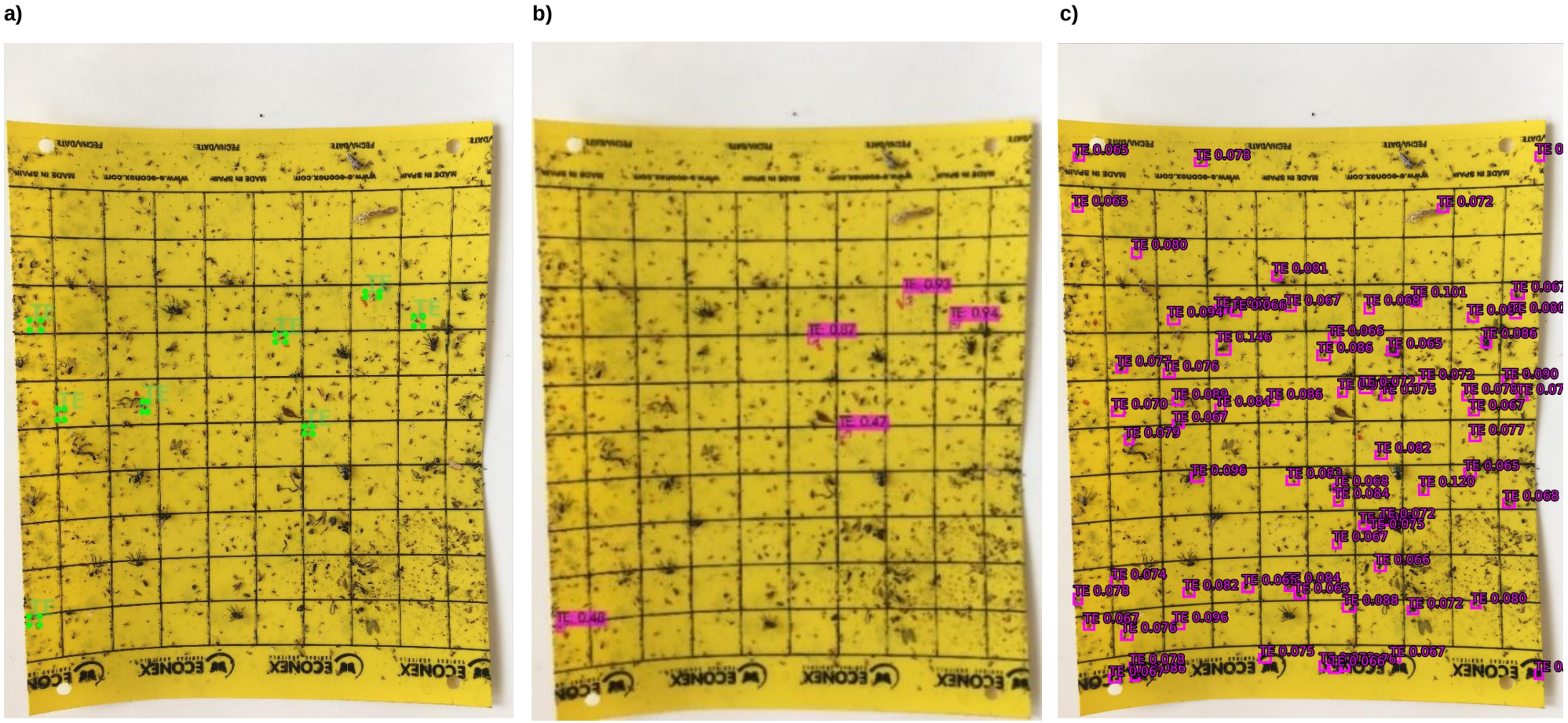
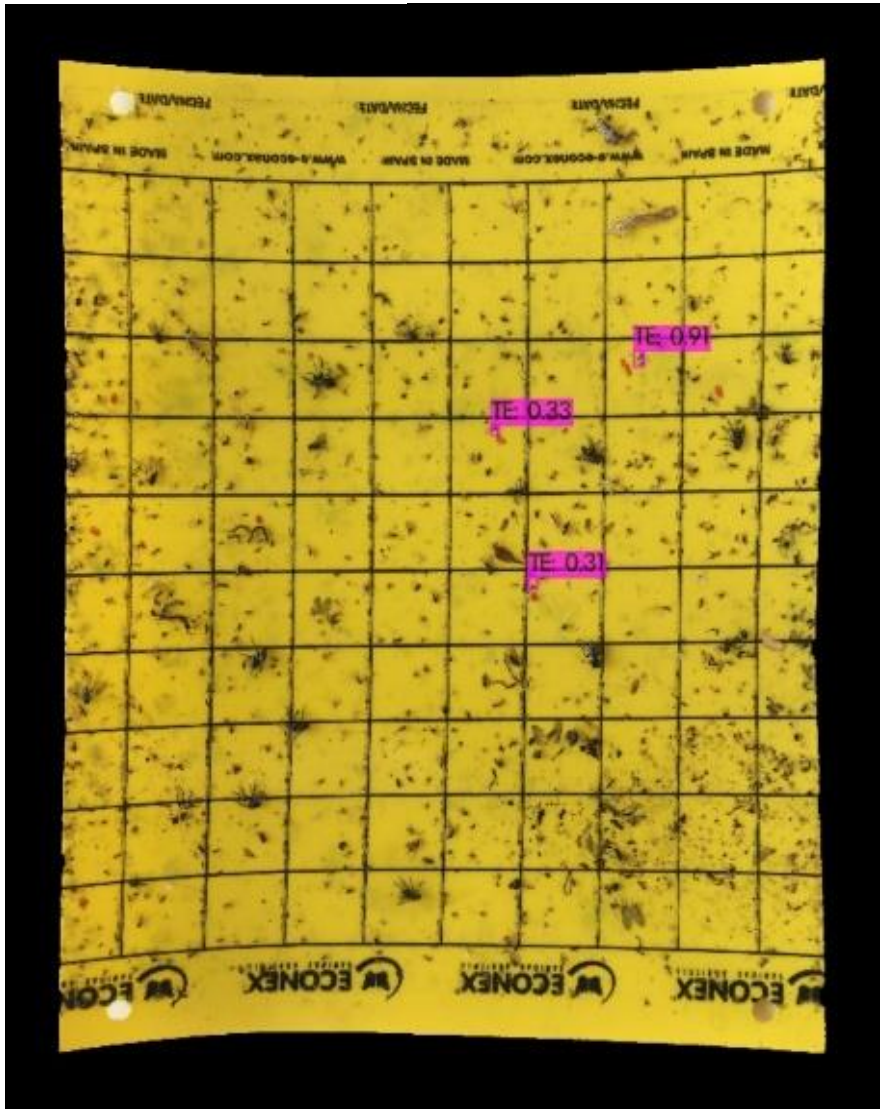


Fig. 36 - Detection of TE of models trained with unprocessed data. a) Ground-truth labeled images. The green dots represent the bounding boxes' corners b) YOLOv4's predictions with confidence scores. c) FRCNN's predictions with confidence scores. The magenta color displays predictions of TE and green of ST.

a)



b)

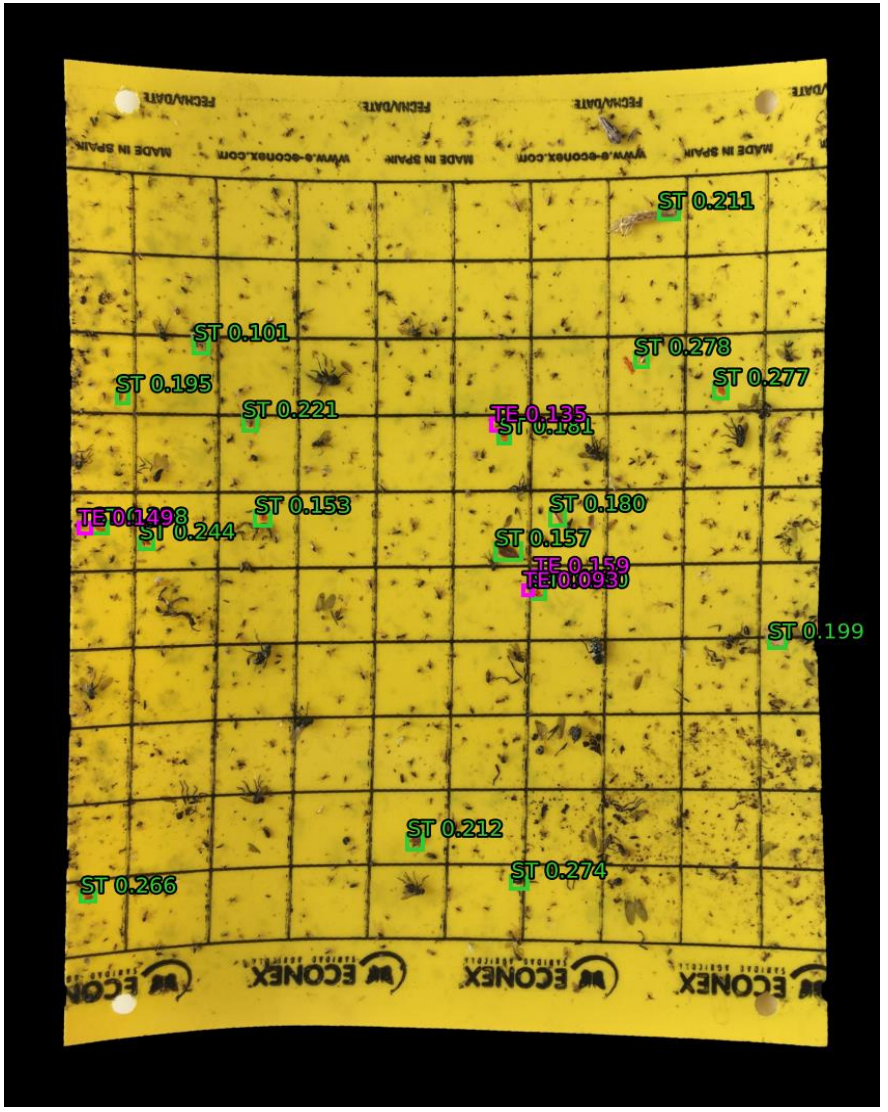


Fig. 37 - Predictions for a processed image. Magenta represents TE and green ST. a) YOLOv4's predictions with confidence scores. b) FRCNN's predictions with confidence scores. The magenta color displays predictions of TE and green of ST.

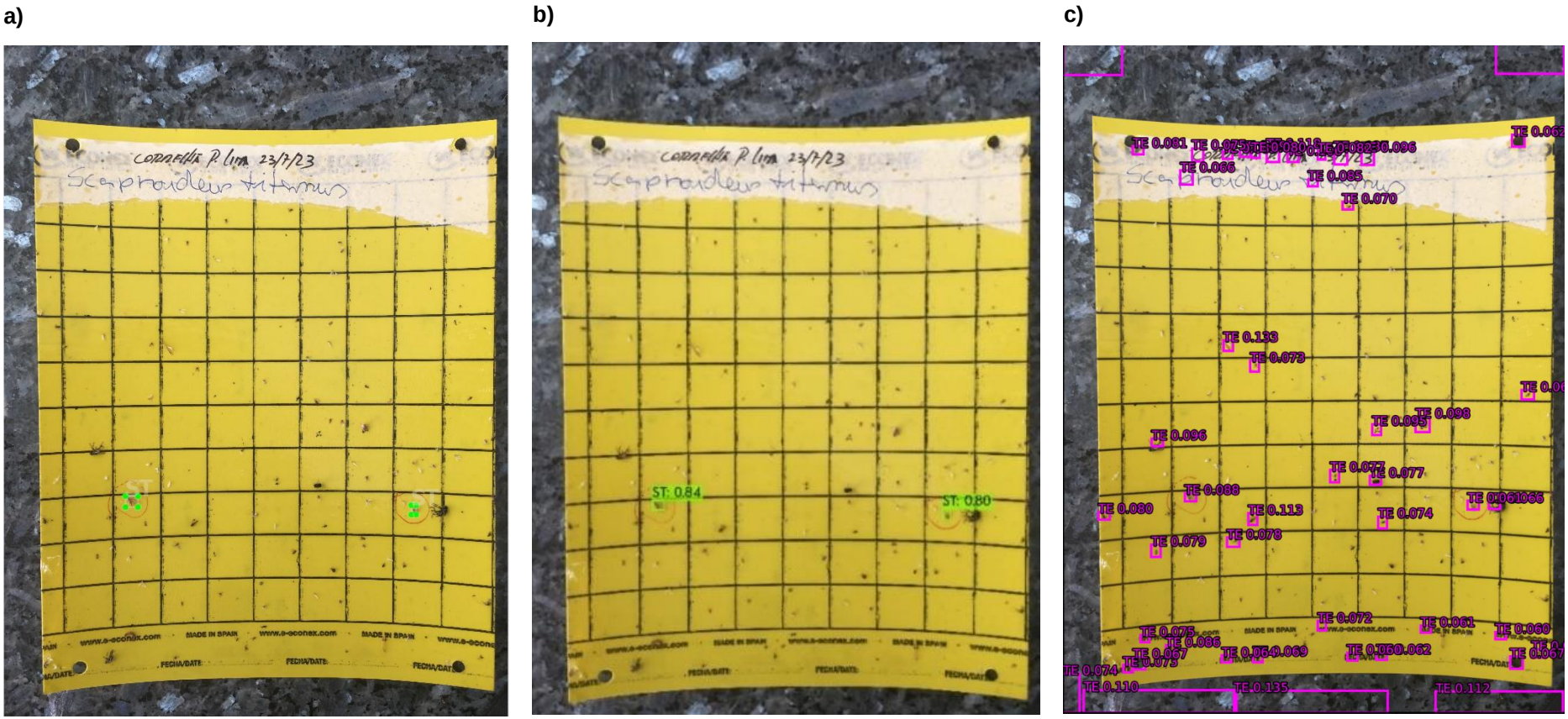


Fig. 38 - Detection of ST of models trained with unprocessed images. a) Ground-truth labeled image. The green dots represent the bounding boxes' corners. b) YOLOv4's prediction with confidence scores. c) FRCNN's predictions with confidence scores. The magenta color displays predictions of TE and green of ST.

a)

b)

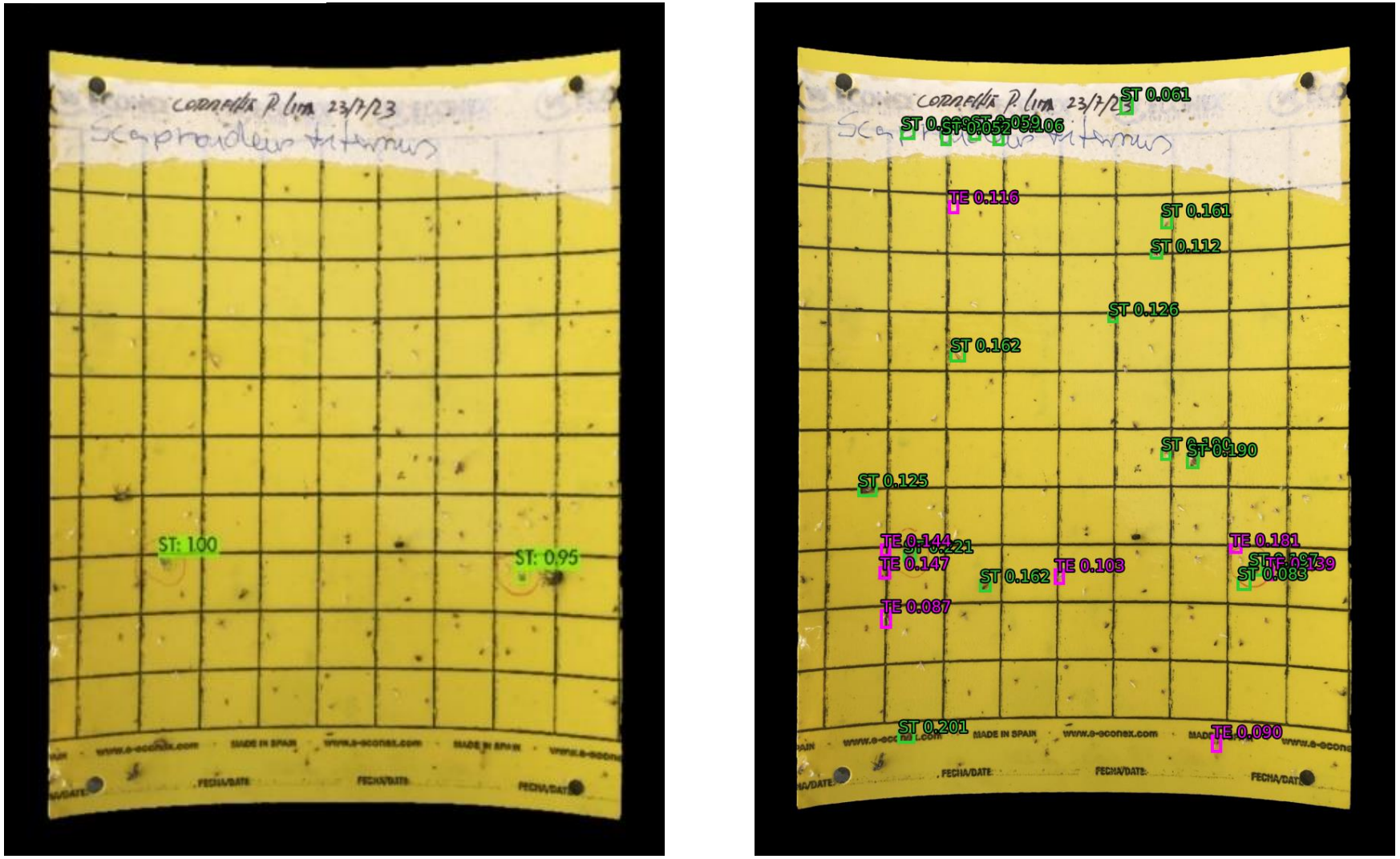


Fig. 39 - Detection of Scaphoideus titanus of models trained with processed images. a) YOLOv4's prediction with confidence scores. b) FRCNN's predictions with confidence scores. The magenta color displays predictions of TE and green of ST.

6. Conclusion

HLB and FD cause significant losses to agriculture. The early identification of their vectors is crucial to avoid dissemination, particularly of TE, since HLB has not been reported in Europe. Portugal and Spain, who have disclosed the presence this pest on their territories, can become gateways to a wave of production loss in the European citrus industry. The current entomological identification process for both insects is time-consuming and inefficient and would greatly benefit from an automatic and user-friendly method that would also be accurate.

To develop a functional automatic process for identification of TE and ST, it is crucial that a larger amount of data is collected. Also, if the method developed during this work would to be deployed, certain trap conditions should be prioritized – avoid using plastic around the trap and yellow objects, like straps, avoid causing pressure that would distort the trap's shape, avoid angles that might cast a shadow or cause heavy specular reflection, try to maintain the trap parallel to the camera, and if there is a need to annotate the trap, it is preferable to either write directly on it with a marker, or use a paper tag but avoid sticking it close to corners or margins.

The segmentation process is very sensitive to the issues mentioned. Implementing a more robust method – such as Mask R-CNN or U-Net – or improve the one employed can better corner detection and, consequently, geometric transformations. To enhance the latter, the number of control and moving points should be increased, or, instead of second-degree polynomial transformation, an affine transformation should be tested.

The object detection models did not achieve results to replace the current identification process. However, the pre-trained YOLOv4 models have demonstrated potential if given enough data. Since the models trained with processed images underperformed, the processing methods, as mentioned previously, should be improved.

References

Aglave, B., 2019. Handbook of plant disease identification and management. CRC Press, Taylor & Francis Group, Boca Raton.

Aguin-Pombo, D., Aguiar, A.M.F., Cravo, D., 2020. First report of Scaphoideus titanus for Madeira Island. EPPO Bull 50, 564–567. https://doi.org/10.1111/epp.12683

Aidoo, O.F., Tanga, C.M., Paris, T.M., Allan, S.A., Mohamed, S.A., Khamis, F.M., Sétamou, M., Borgemeister, C., Ekesi, S., 2019. Size and shape analysis of *Trioza erytreae* Del Guercio (Hemiptera: Triozidae), vector of citrus huanglongbing disease: Size and shape analysis of *Trioza erytreae* Del Guercio (Hemiptera: Triozidae). Pest. Manag. Sci. 75, 760–771. https://doi.org/10.1002/ps.5176

Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET). Presented at the 2017 International Conference on Engineering and Technology (ICET), IEEE, Antalya, pp. 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Andrew Ng, 2016. Nuts and Bolts of Building Applications using Deep Learning.

Bhargava, A., Bansal, A., 2021. Fruits and vegetables quality evaluation using computer vision: A review. Journal of King Saud University - Computer and Information Sciences 33, 243–257. https://doi.org/10.1016/j.jksuci.2018.06.002

Bochkovskiy, A., 2020. AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet) [WWW Document]. Github. URL https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects (accessed 9.12.21).

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs, eess].

Bové, J.M., 2006. HUANGLONGBING: A DESTRUCTIVE, NEWLY-EMERGING, CENTURY-OLD DISEASE OF CITRUS. Journal of Plant Pathology 88, 7–37.

Buduma, N., Locascio, N., 2017. Fundamentals of deep learning: designing next-generation machine intelligence algorithms, First edition. ed. O'Reilly Media, Sebastopol, CA.

Caudwell, A., 1990. Epidemiology and characterization of Flavescence dorée (FD) and other grapevine yellows. Agronomie 10, 655–663. https://doi.org/10.1051/agro:19900806

Center for Invasive Species and Ecosystem Health, 2018. IPM Images: The Source for Agriculture and Pest Management Pictures | Ag Data Commons [WWW Document]. Ag Data Commons. URL https://data.nal.usda.gov/dataset/ipm-images-source-agriculture-and-pest-management-pictures (accessed 9.13.21).

Chen, L.-L., Chung, W.-C., Lin, C.-P., Kuo, C.-H., 2012. Comparative Analysis of Gene Content Evolution in Phytoplasmas and Mycoplasmas. PLoS ONE 7, e34407. https://doi.org/10.1371/journal.pone.0034407

Chouhan, S.S., Singh, U.P., Jain, S., 2020. Applications of Computer Vision in Plant Pathology: A Survey. Arch Computat Methods Eng 27, 611–632. https://doi.org/10.1007/s11831-019-09324-0

Chuche, J., Thiéry, D., 2014. Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: a review. Agron. Sustain. Dev. 34, 381–403. https://doi.org/10.1007/s13593-014-0208-7

Cocuzza, G.E.M., Alberto, U., Hernández-Suárez, E., Siverio, F., Di Silvestro, S., Tena, A., Carmelo, R., 2017. A review on Trioza erytreae (African citrus psyllid), now

in mainland Europe, and its potential risk as vector of huanglongbing (HLB) in citrus. J Pest Sci 90, 1–17. https://doi.org/10.1007/s10340-016-0804-1

Dala-Paula, B.M., Plotto, A., Bai, J., Manthey, J.A., Baldwin, E.A., Ferrarezi, R.S., Gloria, M.B.A., 2019. Effect of Huanglongbing or Greening Disease on Orange Juice Quality, a Review. Front. Plant Sci. 9. https://doi.org/10.3389/fpls.2018.01976

Dermastia, M., Bertaccini, A., Constable, F., Mehle, N., 2017. Grapevine Yellows Diseases and Their Phytoplasma Agents: Biology and Detection, SpringerBriefs in Agriculture. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-50648-7

DGAV, 2018. Flavescência dourada. Situação atual da doença no território nacional.

DRAPCentro, 2017. Psica Africana dos Citrinos. Trioza erytreae (Del Guercio). Ficha Técnica. Governo de Portugal.

EPPO Global Database, 2021. Grapevine flavescence dorée phytoplasma (PHYP64)[World distribution] [WWW Document]. EPPO Global Database. URL https://gd.eppo.int/taxon/PHYP64/distribution (accessed 9.9.21).

EPPO Global Database, 1995. First report of Toxoptera citricidus and Trioza erytreae in Madeira (PT) (No. 1995/07), EPPO Reporting Service no. 01 - 1995.

Folimonova, S.Y., Achor, D.S., 2010. Early Events of Citrus Greening (Huanglongbing) Disease Development at the Ultrastructural Level. Phytopathology® 100, 949–958. https://doi.org/10.1094/PHYTO-100-9-0949

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybernetics 36, 193–202. https://doi.org/10.1007/BF00344251

Gandhi, R., 2018. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. URL https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Gaster, B. (Ed.), 2013. Heterogeneous computing with OpenCL, 2nd ed. ed. Morgan Kaufmann, Waltham, MA.

Ghiasi, G., Lin, T.-Y., Le, Q.V., 2018. DropBlock: A regularization method for convolutional networks. arXiv:1810.12890 [cs].

Girshick, R., 2015. Fast R-CNN. arXiv:1504.08083 [cs].

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, USA, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81

Gonzalez, R.C., Woods, R.E., 2008. Digital image processing, 3rd ed. ed. Prentice Hall, Upper Saddle River, N.J.

Halbert, S.E., Manjunath, K.L., 2004. ASIAN CITRUS PSYLLIDS (STERNORRHYNCHA: PSYLLIDAE) AND GREENING DISEASE OF CITRUS: A LITERATURE REVIEW AND ASSESSMENT OF RISK IN FLORIDA. Florida Entomologist 87, 330–353. https://doi.org/10.1653/0015-4040(2004)087[0330:ACPSPA]2.0.CO;2

Hao, W., Zhili, S., 2020. Improved Mosaic: Algorithms for more Complex Images. J. Phys.: Conf. Ser. 1684, 012094. https://doi.org/10.1088/1742-6596/1684/1/012094

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer

Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. arXiv:1406.4729 [cs] 8691, 346–361. https://doi.org/10.1007/978-3-319-10578-9_23

Hemming, J., Rath, T., 2001. PA—Precision Agriculture: Computer-Vision-based Weed Identification under Field Conditions using Controlled Lighting. Journal of Agricultural Engineering Research 78, 233–243. https://doi.org/10.1006/jaer.2000.0639

Hernández, A.G., 2003. Trioza erytreae (Del Guercio 1918) nueva plaga de los cítricos en Canarias. Phytoma España: La revista profesional de sanidad vegetal 112–118.

Hogenhout, S.A., 2009. Plant Pathogens, Minor (Phytoplasmas), in: Encyclopedia of Microbiology. Elsevier, pp. 678–688. https://doi.org/10.1016/B978-012373944-5.00348-5

Høye, T.T., Ärje, J., Bjerge, K., Hansen, O.L.P., Iosifidis, A., Leese, F., Mann, H.M.R., Meissner, K., Melvad, C., Raitoharju, J., 2021. Deep learning and computer vision will transform entomology. Proc Natl Acad Sci USA 118, e2002545117. https://doi.org/10.1073/pnas.2002545117

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2018. Densely Connected Convolutional Networks. arXiv:1608.06993 [cs].

Huang, Z., Wang, J., 2019. DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection. arXiv:1903.08589 [cs].

Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology 160, 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837

Hubel, D.H., Wiesel, T.N., 1959. RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX. Journal of Physiology 148, 574–591.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167 [cs].

Jacob Solawetz, 2020. Breaking Down YOLOv4 [WWW Document]. Roboflow Blog. URL https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/ (accessed 9.5.21).

Jocher, G., Stoken, A., Borovec, J., NanoCode012, Chaurasia, A., TaoXie, Changyu, L., V, A., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, Hajek, J., Diaconu, L., Marc, Kwon, Y., oleg, wanghaoyang0106, Defretin, Y., Lohia, A., ml5ah, Milanko, B., Fineran, B., Khromov, D., Yiwei, D., Doug, Durgesh, Ingham, F., 2021. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations. Zenodo. https://doi.org/10.5281/zenodo.4679653

Johnson, E.G., Wu, J., Bright, D.B., Graham, J.H., 2014. Association of ' *Candidatus* Liberibacter asiaticus' root infection, but not phloem plugging with root loss on huanglongbing-affected trees prior to appearance of foliar symptoms. Plant Pathol 63, 290–298. https://doi.org/10.1111/ppa.12109

Joseph Redmon, 2020. I stopped doing CV research because I saw the impact my work was having. I loved the work but the military applications and privacy concerns eventually became impossible to ignore. https://t.co/DMa6evaQZr. @pjreddie. URL https://twitter.com/pjreddie/status/1230524770350817280 (accessed 8.29.21).

Kar, K., 2020. Mastering computer vision with TensorFlow 2.x: build advanced computer vision applications using machine learning and deep learning techniques.

Kasper-Eulaers, M., Hahn, N., Berger, S., Sebulonsen, T., Myrland, Ø., Kummervold, P.E., 2021. Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5. Algorithms 14, 114. https://doi.org/10.3390/a14040114

Ker, J., Wang, L., Rao, J., Lim, T., 2018. Deep Learning Applications in Medical Image Analysis. IEEE Access 6, 9375–9389. https://doi.org/10.1109/ACCESS.2017.2788044

Kim, K., Hyun, J., Kim, H., Lim, H., Myung, H., 2019. A Deep Learning-Based Automatic Mosquito Sensing and Control System for Urban Mosquito Habitats. Sensors 19, 2785. https://doi.org/10.3390/s19122785

Kim, P., 2017. MATLAB deep learning: with machine learning, neural networks and artificial intelligence. Apress, New York, NY.

Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S., 2021. A Survey of Deep Learning Applications to Autonomous Vehicle Control. IEEE Trans. Intell. Transport. Syst. 22, 712–733. https://doi.org/10.1109/TITS.2019.2962338

Kuznetsova, A., Maleva, T., Soloviev, V., 2020. Detecting Apples in Orchards Using YOLOv3 and YOLOv5 in General and Close-Up Images, in: Han, M., Qin, S., Zhang, N. (Eds.), Advances in Neural Networks – ISNN 2020, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 233–243. https://doi.org/10.1007/978-3-030-64221-1_20

Lessio, F., Alma, A., 2004. Dispersal patterns and chromatic response of Scaphoideus titanus Ball (Homoptera Cicadellidae), vector of the phytoplasma agent of grapevine flavescence doree. Agric Forest Ent 6, 121–128. https://doi.org/10.1111/j.1461-9563.2004.00212.x

Li, F.-F., Justin Johnson, Serena Yeung, 2017. Lecture 11: Detection and Segmentation.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature Pyramid Networks for Object Detection. arXiv:1612.03144 [cs].

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical Image Analysis 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path Aggregation Network for Instance Segmentation. arXiv:1803.01534 [cs].

Liu, Y., 2020. Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition. World Scientific Research Journal 276–284.

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E., Wen, S., 2020. PP-YOLO: An Effective and Efficient Implementation of Object Detector. arXiv:2007.12099 [cs].

Loshchilov, I., Hutter, F., 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983 [cs, math].

Maixner, M., 1993. Scaphoideus titanus a Possible Vector of Grapevine Yellows in New York. Plant Dis. 77, 408. https://doi.org/10.1094/PD-77-0408

Martini, M., Murari, E., Mori, N., Bertaccini, A., 1999. Identification and Epidemic Distribution of Two *Flavescence Dorée* —Related Phytoplasmas in Veneto (Italy). Plant Disease 83, 925–930. https://doi.org/10.1094/PDIS.1999.83.10.925

Masi, I., Wu, Y., Hassner, T., Natarajan, P., 2018. Deep Face Recognition: A Survey, in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). Presented at the 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, Parana, pp. 471–478. https://doi.org/10.1109/SIBGRAPI.2018.00067

Mazzoni, V., Trona, F., Ioriatti, C., Lucchi, A., Eriksson, A., Anfora, G., 2011. Attractiveness of different colours to Scaphoideus titanus Ball (Hemiptera: Cicadellidae) adults. IOBC/ wprs Bulletin 67, 5.

Mcculloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology 5, 115–133.

Morone, C., Boveri, M., Giosuè, S., Gotta, P., Rossi, V., Scapin, I., Marzachì, C., 2007. Epidemiology of Flavescence Dorée in Vineyards in Northwestern Italy. Phytopathology® 97, 1422–1427. https://doi.org/10.1094/PHYTO-97-11-1422

Murray, R.G.E., Stackebrandt, E., 1995. Taxonomic Note: Implementation of the Provisional Status Candidatus for Incompletely Described Procaryotes. International Journal of Systematic Bacteriology 45, 186–187. https://doi.org/10.1099/00207713-45-1-186

Nelson, J., JUN 12, J.S., Read, 2020 16 Min, 2020. Responding to the Controversy about YOLOv5 [WWW Document]. Roboflow Blog. URL https://blog.roboflow.com/yolov4-versus-yolov5/ (accessed 8.29.21).

Nieuwenhuizen, A., Hemming, J., Suh, H.K., 2018. Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN.

Oommen, P., 2020. ResNets — Residual Blocks & Deep Residual Learning [WWW Document]. Medium. URL https://towardsdatascience.com/resnets-residual-blocks-deep-residual-learning-a231a0ee73d2 (accessed 9.10.21).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Pytorch. https://doi.org/1912.01703

Patrício, D.I., Rieder, R., 2018. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and Electronics in Agriculture 153, 69–81. https://doi.org/10.1016/j.compag.2018.08.001

Pérez-Otero, R., Mansilla, J.P., del Estal, P., 2015. Detección de la psila africana de los cítricos, Trioza erytreae (Del Guercio, 1918) (Hemiptera: Psylloidea: Triozidae), en la Península Ibérica. ARQUIVOS ENTOMOLÓXICOS 119–122.

Quartau, J.A., Guimarães, J.M., André, G., 2001. On the occurrence in Portugal of the nearctic Scaphoideus titanus Ball (Homoptera, Cicadellidae), the natural vector of the grapevine “Flavescence dorée” (FD). IOBC wprs Bulletin, Integrated Control in Viticulture 24, 273–276.

Raschka, S., Mirjalili, V., 2019. Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow 2.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs].

Redmon, J., Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger. arXiv:1612.08242 [cs].

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497 [cs].

Ribani, R., Marengoni, M., 2019. A Survey of Transfer Learning for Convolutional Neural Networks, in: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T). Presented at the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), IEEE,

Rio de Janeiro, Brazil, pp. 47–57. https://doi.org/10.1109/SIBGRAPI-T.2019.00010

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65, 386–408. https://doi.org/10.1037/h0042519

Samways, M.J., 1987. Phototactic response of *Trioza erytreae* (Del Guercio) (Hemiptera: Triozidae) to yellow-coloured surfaces, and an attempt at commercial suppression using yellow barriers and trap trees. Bull. Entomol. Res. 77, 91–98. https://doi.org/10.1017/S0007485300011573

Sejnowski, T.J., 2018. The deep learning revolution. The MIT Press, Cambridge, Massachusetts.

Shilpa Ananth, 2019. Faster R-CNN for object detection. A technical paper summary. towards data science. URL https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46

Shin, H.-C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging 35, 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0

Stackebrandt, E., Frederiksen, W., Garrity, G.M., Grimont, P.A.D., Kämpfer, P., Maiden, M.C.J., Nesme, X., Rosselló-Mora, R., Swings, J., Trüper, H.G., Vauterin, L., Ward, A.C., Whitman, W.B., 2002. Report of the ad hoc committee for the re-evaluation of the species definition in bacteriology. Int J Syst Evol Microbiol 52, 1043–1047. https://doi.org/10.1099/00207713-52-3-1043

Sugio, A., Kingdom, H.N., MacLean, A.M., Grieve, V.M., Hogenhout, S.A., 2011. Phytoplasma protein effector SAP11 enhances insect vector reproduction by manipulating plant development and defense hormone biosynthesis. Proceedings of the National Academy of Sciences 108, E1254–E1263. https://doi.org/10.1073/pnas.1105664108

Supeshala, C., 2020. YOLO v4 or YOLO v5 or PP-YOLO? [WWW Document]. Medium. URL https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109 (accessed 8.29.21).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the Inception Architecture for Computer Vision.

Tamesse, J.L., Messi, J., 2002. Incidence de Trioza erytreae (del Guercio) (Homoptera: Triozidae), Psylle Vecteur du Greening sur la Sensibilité des Plantules d'Agrumes dans une Pépinière au Cameroun. Int. J. Trop. Insect Sci. 22, 97–103. https://doi.org/10.1017/S1742758400015174

Tan, M., Pang, R., Le, Q.V., 2020. EfficientDet: Scalable and Efficient Object Detection. arXiv:1911.09070 [cs, eess].

Tan, R.J., 2019. Breaking down Mean Average Precision (mAP) [WWW Document]. Medium. URL https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52 (accessed 9.20.21).

Tramontini, S., Delbianco, A., Vos, S., 2020. Pest survey card on flavescence dorée phytoplasma and its vector Scaphoideus titanus. EFS3 17. https://doi.org/10.2903/sp.efsa.2020.EN-1909

Tzutalin, 2015. LabelImg [WWW Document]. Github. URL https://github.com/tzutalin/labelImg (accessed 9.25.21).

Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M., 2013. Selective Search for Object Recognition. Int J Comput Vis 104, 154–171. https://doi.org/10.1007/s11263-013-0620-5

van den Berg, M.A., Deacon, V.E., Thomas, C.D., 1991. Ecology of the citrus psylla, Trioza erytreae (Hemiptera: Triozidae). 4. Settling and general behaviour of nymphs. Phytophylactica 201–208.

Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., 2018. Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience 2018, 1–13. https://doi.org/10.1155/2018/7068349

Wang, C.-Y., Liao, H.-Y.M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., 2019. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. arXiv:1911.11929 [cs].

Wang, N., 2019. The Citrus Huanglongbing Crisis and Potential Solutions. Molecular Plant 12, 607–609. https://doi.org/10.1016/j.molp.2019.03.008

Whitley, D., 1994. A genetic algorithm tutorial. Stat Comput 4, 65–85. https://doi.org/10.1007/BF00175354

Woo, S., Park, J., Lee, J.-Y., Kweon, I.S., 2018. CBAM: Convolutional Block Attention Module. arXiv:1807.06521 [cs].

Yao, Z., Cao, Y., Zheng, S., Huang, G., Lin, S., 2020. Cross-Iteration Batch Normalization. arXiv:2002.05712 [cs, stat].

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y., 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. arXiv:1905.04899 [cs].

Zaidi, N.I.B., Lokman, N.A.A.B., Daud, M.R., Achmad, M.S.H., Khor, A.C., 2015. FIRE RECOGNITION USING RGB AND YCBCR COLOR SPACE. ARPN Journal of Engineering and Applied Sciences 5.

Zhang, E., Zhang, Y., 2009a. Precision, in: LIU, L., ÖZSU, M.T. (Eds.), Encyclopedia of Database Systems. Springer US, Boston, MA, pp. 2126–2126. https://doi.org/10.1007/978-0-387-39940-9_480

Zhang, E., Zhang, Y., 2009b. Recall, in: LIU, L., ÖZSU, M.T. (Eds.), Encyclopedia of Database Systems. Springer US, Boston, MA, pp. 2348–2348. https://doi.org/10.1007/978-0-387-39940-9_479

Zhang, E., Zhang, Y., 2009c. Average Precision, in: LIU, L., ÖZSU, M.T. (Eds.), Encyclopedia of Database Systems. Springer US, Boston, MA, pp. 192–193. https://doi.org/10.1007/978-0-387-39940-9_482

Zhang, W.J., Yang, G., Lin, Y., Ji, C., Gupta, M.M., 2018. On Definition of Deep Learning, in: 2018 World Automation Congress (WAC). Presented at the 2018 World Automation Congress (WAC), IEEE, Stevenson, WA, pp. 1–5. https://doi.org/10.23919/WAC.2018.8430387

Zhang, Zhi, He, T., Zhang, H., Zhang, Zhongyue, Xie, J., Li, M., 2019. Bag of Freebies for Training Object Detection Neural Networks. arXiv:1902.04103 [cs].

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D., 2019. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. arXiv:1911.08287 [cs].

Zhong, Y., Gao, J., Lei, Q., Zhou, Y., 2018. A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture. Sensors (Basel) 18, 1489. https://doi.org/10.3390/s18051489