VINBOT: robot vision and learning from vineyards Automatic segmentation for yield estimation

João Mak Duarte
joaomakduarte@tecnico.ulisboa.pt
Instituto Superior Técnico

***Abstract*—The goal of this work is to estimate a vineyard's yield based on the visible area of grape bunches resulting from an autonomous segmentation in a set of images of a vineyard.**

Firstly the problem of autonomous segmentation is tackled by the use of a FCN, trained with data from the ISA vineyard with minor data augmentation, operation which increases the number of images. The FCN is tested and its loss function adjusted to compensate the imbalance present in the data set, where the grape clusters only represent 3.5% of the entire images. This is complemented by pre and post processing operations that improve the segmentation's score, the IOU. This metric evaluates how well the segmentation overlaps the ground truth. The pre processing is composed by a sliding window and a colour space change that increased the test set score to 62%. As for the post processing, the morphological operation "open" is used and the image rebuilt with the objective of removing false positives. The combination of these efforts result in a IOU score of 64%.

In the second part of testing, the yield is estimated with the use of two models, one that predicts the percentage of grape bunches hidden in the image according to the porosity of the vine, and another that transforms the total area of bunches into volume. Four different cases are presented, two varieties, encruzado and arinto, from the same to stages, harvest and veraison. The veraison results achieve the desired metric score of an error less than 10% for both varieties, 3% for encruzado and just under 10% for arinto. Although some aspects of the overall process need improvement, in order to make it more robust, the results were satisfactory for this part.

***Index Terms*—Computer vision, Precision Viticulture, Yield estimation, Machine learning**

I. Introduction

Since automation of systems began to be a standard in every area of human production, it has increased crop output in agriculture, reduced manual labour and created an overall improvement in quality of life, according to [1]. Also, the more frequent use of robotics in agriculture is due to the lack of human resources relative to manual labour and to the increasing business competitiveness.

These developments include harvesting, seeding, irrigation and other type of robots related to the basics of agriculture activity. Along with the new possibilities that technology brings, also new strategies have been developed in relation to agriculture, one of which being Precision Agriculture(PA). PA has been the trend in most crops. The idea is that each parcel of field is different, and as such, should have different needs. With PA, new challenges are brought to attention. This more detailed information over a field, in this particular case a vineyard, allows for a more precise control over the plantation over more precise techniques related to yield estimation, quality analysis or post harvest production.

The first and foremost issue with yield estimation is the vine's natural variability. Any vine may give significantly different yields depending on the year (temporal variability), soil or weather conditions, biotic or abiotic stresses, variety or agriculture practices [2], [3]. Given the difficulties exposed and with the advancements of robotics, especially sensor-based technology, some works have been made in order to be able to develop an automated system of yield estimation. One of the paths pursued is the use of computer vision. Having images as data, Machine Learning(ML) applied to image processing is one of the most common trends [4]–[7].

These activities have taken a prominent importance in the current agriculture research. That being said, the study of an accurate yield estimation system, regardless of the crop in question, has increasingly become a necessity. In the specific case of viticulture, an accurate yield estimation brings significant advantages such as: correct estimation of cellar needs, the possibility of developing targeted marketing strategies, knowing in advance the amount of machinery and manpower needed for harvest, allocating cellar space and equipment and managing stock prices for both the grapes and the produced wine [8].

In ML, with the development of processing units and new open-sourced programming libraries, the difficulty of applying not only classical methods, such as statistical models, but Neural Networks(NN) as well, has decreased.

A. Problem formulation

ISA is developing a project with the goal of estimating a vine's yield without invasive operations. This project is based on a moving robot with sensors that collects data along their vineyard's lines. At the moment, no part of this process is automatic. That being said, the main goal of this work is to create an algorithm that has as an output the visible area of grape clusters in any given image provided by the mobile platform, in order to automatise this step of the process. This first step in automation has additional problems other than the main one of creating a system to replace the hand segmentation.

B. Outline

In Chapter II, the main practises in precision viticulture referring to yield estimation will be presented, alongside a

few projects that also relate ML to the problem and a brief review of image segmentation techniques.

In Chapter III, the system that will be used in solving the problem is presented and described in detail, starting with the necessary image pre-processing, passing through the grape bunch segmentation, followed by the prediction post-processing and yield estimation.

In Chapter IV, experiments are made on the encruzado variety in order to better understand which variables of the system are best for the task. The results analysed and discussed.

Finally in Chapter V, tasks and ideas that could improve the overall of this process and provide continuity are proposed and the project's conclusions are shown.

II. Related Work

A. Yield estimation

As a generalised practise [9], the way to estimate yield requires a deep knowledge of the vineyard variability in space and time, combined with years of expertise in viticulture. Firstly, a set of samples is taken from several parts of the vineyard where the producer knows to be different from one another. Following the sampling, the producer weighs the set and extrapolates for the patch where it was taken from and, finally, the estimates are added, resulting in the final prediction. The way the extrapolation is made varies from producer to producer since it also takes into consideration empirical knowledge. These sort of methods are time consuming and can be destructive to the crops.

For PA, yield estimation has been not only a commodity but a necessity. For this specific problem, new alternative methods have been developed and used commercially. Usually these methods present some limitations, the main one being that they rely on invasive techniques such as defoliation, as shown in Figure 1, or that they are aimed at yield estimation at a larger scale.

Fig. 1: Example of a defoliated vine

Given the utmost importance of being able to estimate the yield , several efforts have been made to develop new strategies and technology that support this area. Some methods are already in use, like the aeropalynological forecast models [10], although this is more directed at a regional scale production estimate. This method is based on vineyard pollen readings and correlates the amount of pollen concentration in the air of a certain region which increases with the number of flowers and, consequently, the number of future grapes. So, the current trend has been in sensor based technology. Another method that is still under development is the one proposed by [11], where is said the tensile strength of the vineyard's supporting wires is adjusted to be proportional to the weight of the existing bunches. This method has the limitation of requiring large investments in sensors that will also require regular maintenance. Although quite a few different approaches have been made, the main trend has been visual-based methods. There are some currently under development and others already tested.

B. Computer vision in the viticulture and agronomic context

Computer vision has become one of the most common strategies adopted in the yield estimation problem, not only regarding bunch recognition [8], [12], but also shoots [13], flowers [14], [15] and berries [16], [17]. Using computer vision, it is reasonable to assume that a more correct identification of the yield components(bunches, berries, flowers or shoots) will provide a more accurate estimation.

C. Image Segmentation

Consequently, looking at reviews from the recent years, the main work effort has been oriented towards computer vision, more specifically in ML [18]–[20]. In particular, NN have demonstrated to outdo expectations in several fields, in particular the FCN, presented in [21], when trained end-to-end, pixels-to-pixels on semantic segmentation exceed the previous best results without further machinery. Another approach to this problem is the technique of transfer learning. Having the problem of lack of data, it is possible to adapt a existing pre-trained classifying NN, changing the fully connected layers into a deconvolution, as proposed by [21]. This utilises the feature learning part of the network, being that the only trained part is the deconvolution or upsampling part, with the data specific to the problem(Figure 2). This structure is further explained in Chapter III.

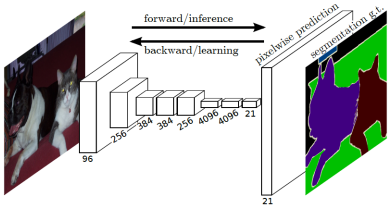


Fig. 2: FCN architecture
(Source: [21])

Of course the limitations to any NN solution revolve around the lack of direct control over the classification process and feature learning, the possibility of over fitting and the inability to be certain when a minimum of the loss function is reached, that it is the global minimum that will solve the problem optimally. Also, the amount of data that is usually required is significantly more than the amount used for traditional image segmentation methods.

Another important work done in image segmentation based on Convolutional Neural Networks(CNN) is [22]. The U-Net (Figure 3) proposed, is comprised of two stages: firstly the compression stage that is dedicated to feature extracting resulting in a multi-channel feature map; and in the second stage, it is added a usual contracting network by successive layers, replacing the pooling layers with upsampling operators, increasing the output's resolution. This network provides the possibility of end-to-end training with a reduced data set, obtaining satisfying results.

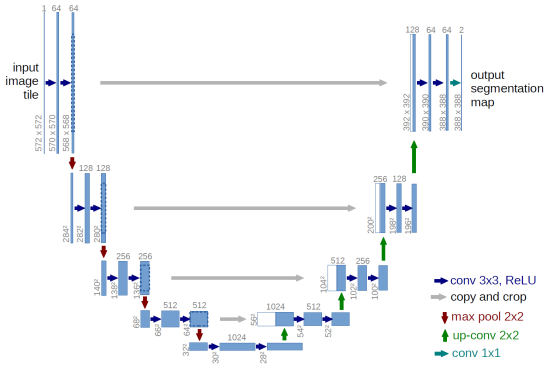


Fig. 3: U-Net architecture
(Source: [22])

Regardless, to opt for a classic approach in segmentation through computer vision would also be an option, with the advantages of having a greater control of the overall process, resulting in an easier feature tweaking, on one hand, but on another this process may be more time consuming and difficult. That being said, some works combine the two approaches in separate steps of the process, like [7], [23], the first using a combination of FCN with the application of a Hough Transform based method and the second using descriptor vectors that combine Histogram of Oriented Gradients(HOG) and Local Binary Pattern (LBP) followed by a Support Vector Machine (SVM). Another work that combines classical methods with more recent ones, is [24]. With the objective of immature green citrus fruit detection, it performs fruit detection through booth Region-Based Convolutional Neural Networks (F-RCNN) and a multi-level Hough circle method.

D. Yield components recognition

Taking a NN approach to this problem, there are already projects who tackle the same problematic with a similar backbone idea [4], [5], [7], [24] that show promising results.

One of the most interesting strategies is described in [5]. In this paper transfer learning is applied. A pre-trained Classifying Neural Network, the Inception-V3 [25] is used as a base for the localisation algorithm. To be able to correctly classify and localise the bunches in the images, the last layer was replaced by what the authors entitled "localisation head". This head takes the information from the second to last layer and uses it to produce an outcome of probability of a certain area in the image having or not a bunch. Through this probability map, a bounding box is created around the areas with the largest probability value. The "localisation head" was trained separately from the remaining network with images labelled with containing bunches or not. For this training, the algorithm split the images into areas of interest and could correctly classify 99% of them as for containing or not grape bunches.

In [4], the last layer of the NN is replaced with a classification layer made by five neurons, one for each possible classification, bunch, wood, pole, leaves and background. This last layer can be described as a *maxpool* layer, in which the classification is given through the most likely probability of the patch in analysis. The algorithm was tested with four different NN: Alexnet [26]; VGG16 and [27]; VGG19 [27]; GoogLeNet [25].

With the different goal of simply counting the amount of bunches in a given image, this strategy was based on a 80x80 pixel sliding window. The 80x80 size was selected from a mean size of a bunch in an image in their specific data set. This window would be then resized and fed to the NN.
The accuracy results for each NN are: Alexnet (81.03%); VGG16 (83.05%); VGG19 (91.52%); GoogLeNet (79.66%).

Another approach to the identification of yield components is the one presented in [7]. Their aim is one that is very similar to the work of this dissertation, with the difference that the data collection is performed during the flowering stage of the grape vine. Their process is divided into two stages: localisation of inflorescences in the image and single flower extraction. The first step is done through the use of a FCN, with a encoder part adapted from the AlexNet [26] and a decoder with only two up-convolutions. This architecture is derived from the U-Net [22]. They train the network with labelled images of the vines, with the classes of inflorescences and not-inflorescences.

Focusing on the inflorescence detection, since they use a process based on the Hough transform for the flower extraction part, 5292 608x608 pixel images were used to train the network. Given the nature of the problem, the detection and localisation of inflorescences results in regions of interest (ROI) and, as such, mean Intersection Over Union (IOU) was used as a quality measure.

The best results were after the 285500^{th} epoch, which resulted in a mean IOU of 87.6%, with a class-specific IOU of 76% for the inflorescence class.

E. Limitations

The previously mentioned methods, with the exception of [7] lack practicality in some sense. For example, part of them used a data set with vineyards that have had some of their leaves removed, as in Figure 1. Also, for the specific goal of this work, that is estimating yield, when recognising bunches in an image, the objective is to segment them in the image, not only count them, as it is done in [4]. The main limitation, in general, is that to base a method of yield prediction on the bunches that exist, the more of them that are occluded, either by other bunches or leaves, the more uncertain the method is going to be. This work aims to overcome these limitations, starting with not being invasive. The data collection does not interfere with the natural vine development or viticulture

practises. Also, with the prediction models developed at ISA, the occlusion problem is handled.

III. Proposed System

A. Materials

The mobile platform used is the Vinbot robot (Figure 4), that has in its mast a Red-Green-Blue(RGB) camera, at an adjusted height so it matches the canopy height, that is used to collect the image data. The robot is controlled by an operator with a controller and collects the data to an external hard drive directly connected to the robot.

Fig. 4: Vinbot robot

The software used for the image labelling and other small tasks in the post processing is the ImageJ software. The programming was done in its entirety in python and the NN training was performed on the Google Colab online platform.

B. System Structure

The complete system is as described in Figure 5. Firstly, the robot passed through the vineyards with the RGB camera and takes approximately 1 meter wide images, meter by meter. These pictures are automatically pre processed, segmented and post processed. After the information is ready to be extracted from the images, this area of grape clusters in pixels is converted into cm2 and then into weight, estimating the yield.

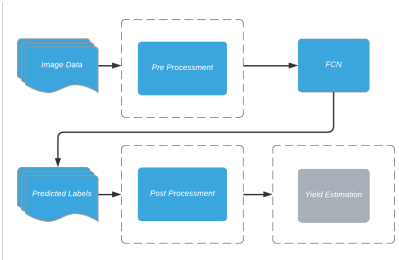


Fig. 5: Complete system overview. In blue are the blocks that concern this work, in grey the system provided by ISA

C. Pre Processing

The objective with pre processing, in this case, is to transform the image data into a format that is more prone to learning, either by accentuating shapes or colours in images, for example so it would be easier for feature learning, or by simply formatting the image so it complies with the specificity of the segmentation algorithm.

The pre processing of the image data can be summarised in the following flowchart (Figure 6):

Fig. 6: Pre Processing operations from the raw image to the processed network's input

1) Colour space: Although RGB is one of the most commonly used models in computer vision [28], it presents some disadvantages. The model produces a nonlinear and discontinues space, which makes the changes in colour hue difficult to pursue. This combined with the fact that the colour hue is also easily affected by illumination changes, makes that colour tracking and analysis a nontrivial task.

That being said, another colour space was considered to replace the RGB model, CIELAB (Figure 7). Firstly, there is a model, RGB normalised, that dealt with one of the major problems, the sensitivity to illumination changes of RGB. The principle that guides this model is that a certain colour is formed using a certain proportion of three primary colours from the model and not a defined amount of each one. However, although it removes the aforementioned negative illumination effect, it also reduces object detection capability, due to the loss of contrast that the same illumination provides [28].

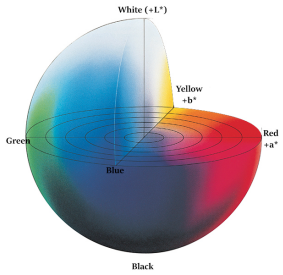


Fig. 7: CIE Lab colour space
source: [29]

The CIELAB model is based primarily on the physics aspect of light. CIELAB is based of the CIE XYZ model. This model is calculated using the light wavelength from the physic representation of any specific colour. The L^* component for this model encapsulates the illumination effect on the colours, providing a way to only remove the unwanted consequences of lighting changes. The CIELAB model can represent colours that are not handled by other models and, theoretically, it could represent an infinite number of chromatic combinations [28].

2) Image formatting: The next step in the system is the actual segmentation of grape clusters in the images. The algorithm selected has as input 572x572 pixel images, but produces an output of 388x388 pixel mask. Due to the nature of the algorithm, the frame pixels(from 388 to 572) will not be classified, since there is a resulting loss in border pixels from the convolutions in the NN. In order to not lose any data two steps were taken.

Firstly, a 388x388 pixel sliding window was passed through the image with minimum overlap, resulting in an image that will have to be enlarged to correctly correspond to the input size of 572x572 pixel. In order to do that a 92 pixel mirror frame was applied, resulting in an image as Figure 8.

Fig. 8: Final image after sliding window and mirror framing

D. Segmentation

The image segmentation step of this system consists of a FCN that will take as input the previously processed images and will have as an output a binary mask of what is, or not, a grape cluster in the provided image. This specific format of NN was chosen due to the fact that its output, in this context, is the area of visible clusters for any specific image. By segmenting the image into bunch and background an area can be calculated by counting the number of pixels classified as bunch. This is a necessity for the final yield estimation model, that takes as one of its inputs an area of visible grape bunches. The architecture chosen for the task was the one used in [22], that will be studied in further detail in this section, along with a brief introduction to FCN. Another issue besides the one presented is that the classes that the networks aims to classify are imbalanced in the data set. In order to correct this imbalance the loss function was adapted.

1) Fully convolutional networks: In general, a FCN can be comprised of two types of layers, convolutional layers and pooling layers.

The convolutional layer is the filter that, when passed through the input, defines what are the feature locations in a feature map. This is the main task of the network [30].

As stated in [31], this type of layer is composed of, essentially, a kernel that slides across the input feature map with a certain stride (distance between two consecutive positions of the kernel). In each position, the dot product is calculated. The resulting products are concatenated producing a new feature map as an output.

Another parameter that can be chosen, is the padding. Padding is the border pixels dimension that can be applied in the convolutional layer that can contribute to altering the size of the output. The output size is determined by Equation 1:

$$o = \frac{i + 2p - k}{s} + 1 \tag{1}$$

Where o is the output size, i the input size, p the padding and k the kernel size.

As for pooling layers, as said in [31], [32], the objective in using them is to reduce space dimension and, consequently, computational power needed to process, to provide invariance to small translations of the input and to minimise overfitting.

The principal behind the pooling layer is from a set of numbers create one that can be representative, according to the desired outcome. For example, one of the most used types of pooling is *max pooling*. Much like a kernel slides on an input map, this operation also has pre defined pooling window size and stride.

As described in [21], the structure of these networks are divided into two parts, the downsampling and upsampling. Both use convolutional layers, although with different purposes. The downsampling part is the only to use pooling layers. Firstly, the downsampling part extracts features from the input, reducing the input size at each layer. The upsampling part is where the final feature map that was calculated, combined with spatial data from the downsampling, reconstructs the input with the new learned information. This reconstruction is made possible through the use of transposed convolutions or up convolutions, which has the advantage of being able to carry out trainable upsampling. It provides as an output a reconstructed input of spatial dimension equal to the input of the correspondent layer in the downsampling part. The combination of these types of layers is the basis of a FCN. Considering this type of network for this system, the U-Net [22] was chosen due to its positive results.

2) U-Net: Considering the existing FCN, the U-Net stood out. It showed positive results obtained with a similar deficiency of training data, which was solved with data augmentation. The U-Net network, as seen in Figure 3, was used in [22] for biomedical image segmentation.

This network is composed by two phases, a contracting and an expansive side, as the FCN described before. In total there are 23 convolutional layers to this network.

3) Loss function: In order for the network to consider the imbalance between classes it needs to have a weight factor in it. Using binary cross entropy just as is, was proven not to be enough, as shown in Chapter IV, due to the fact that the set is highly imbalanced, with a ratio of 3.5% bunch pixels to 96.5% background. As such, it was transformed into a weighted binary cross entropy (Equation 2):

$$H_b(p) = -\frac{1}{N}\sum_{i=1}^{N} w_c(y_i \cdot log(p(y_i))) + w_b((1-y_i)\cdot log(1-p(y_i))) \tag{2}$$

Where y_i is the label(1 or 0) and $p(y_i)$ is the predicted probability of that label. The weights w_c and w_b represent

the weights relative to the class cluster and class background, respectively. It is the objective for the loss function to favour "1" classifications. Therefore, the weight related to the cluster class must be higher than the weight related to the background class.

The values chosen must reflect the nature of the imbalance. Logically the weights should be an inversion of the relevance of each class in the image. Then the weights w_c and w_b will be chosen inversely, with $w_c = 96.5$ and $w_b = 3.5$.

E. Metrics

There will be two types of metrics. In a first stage is important to assess how well the segmentation is made and then, in a later stage, the areas of bunches predicted, after being put through the models that convert them into grape weight, should be rated against the correspondent ground truth. Regarding the first stage, this implies, as stated before, a pixel-wise classification. Therefore, the metric that evaluates the success of the prediction must be one that is not binary but with an associated percentage. That being said, the metric used was the IOU, also known as the Jaccard index(3). It provides a ratio of how much of the predicted class in question is correctly overlapping the corresponding ground truth. The second stage of the evaluation is further explained in Section III-F3.

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{3}$$

F. Post Processing

Given the presence of some noise in the image, i.e. small false positives clusters, post processing operations were made. They removed very small clusters of false positives through the use of morphological operations. Following the operations, the image was reconstructed. In this step, false positives are also eliminated due to the overlapping areas that resulted from the sliding window, which allow for two or more predictions for some part of the image. With the reconstructed image, with the help of a scale in the image, the area of visible bunches in pixels was converted to cm^2.

This being said, the post processing can be divided into three smaller steps (Figure 9).

Fig. 9: Post processing

1) Morphological operations: These operations consist of, firstly, convolving the binary image with a structuring element.

The standard morphological operations are (Figure 10) dilation, erosion, majority, opening and closing.

As it is possible to observe from Figure 10, dilation thickens the object, while erosion thins it. The last two operations, "close" and "open", are a dilation followed by an erosion and the reverse. They tend to leave large regions and smooth boundaries unaffected, while removing small objects or holes

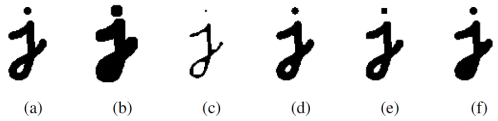


Fig. 10: Examples of morphological operations: (a) original image; (b) dilation; (c) erosion; (d) majority; (e) opening; (f) closing.
source: [33]

and smoothing boundaries. It was these characteristics that made the "open" operation the chosen for the first step of the post processing.

2) Image reconstruction: As stated before, the existence of overlapping areas in the predicted images allows for another mean of dealing with false positives. The way to take advantage of this situation is by creating a decision system for the different areas of overlapping in the reconstructed images regarding what is or not a grape bunch. The image presented in Figure 11 illustrates the overlapping segments, where the number in each segment indicates how many images overlap:

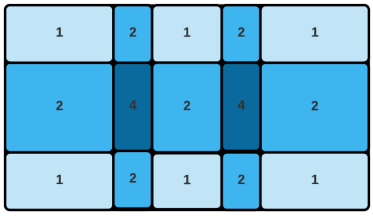


Fig. 11: Regions of overlap in the reconstructed image:"1" is where there is only one sliding window image that classifies that segment; "2" represents that the segment in question is classified by two images;"4" is where the segment is classified by 4 different images

For the "1" regions, the prediction of that image stand for the final reconstruction. For the "2" regions, both predictions must agree on a pixel level, so a *logical and* operation is applied. As for the "4" regions, a 75% certainty is believed to be enough to classify a certain pixel as part of a bunch. Although the problem is an excess of false positives, false negatives also occur. The three out of four approach provides some robustness against false negatives in the reconstruction in opposition to a strategy that is simply based on a *logical and* approach.

3) From counting pixels to yield estimation: After the reconstruction, the following step in the process is to account for the number of pixels classified as grape bunches in each image in order to transform that number into kg. This process starts with calculating the ratio from pixel to cm^2. Since the Vinbot project has not been fully automated, this task is performed by hand, image by image. With the ratio, it is just a matter of transforming the area in pixel into an area in cm^2. These models, developed in [34] and also used in [35], [36], take part in two stages of this process (Figure 12):

The first model is used to estimate the visible bunch percentage of the total existing bunches, taking into consideration

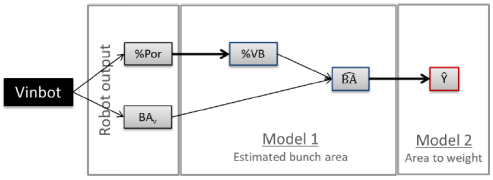


Fig. 12: Yield estimation diagram from the Vinbot output until the final estimation of yield per image. %Por=bunch zone canopy porosity (in percentage); BA_V = visible bunch projected area; %VB = percentage of visible bunches; $\widehat{BA}$ = estimated bunch projected area (total per image); $\hat{Y}$= final yield estimation per image.

the porosity of the vine. The porosity is correlated with the percentage of visible bunches, since the major cause for bunch occlusion is leaves from the vine. Therefore, the higher the porosity percentage (more empty spaces in the canopy) the lesser the leaf occlusion. Knowing the area of visible bunches and the percentage to which it corresponds, due to the porosity correlation, it is possible to make a projection to the total bunch area, visible and covered. This projection is then fed into the second model to calculate the final weight, transforming cm^2 to kg through a polynomial fit where the bunch area is the independent variable and the weight the dependent one.

This second evaluation is important since there are steps in between the output of the segmentation network and the final estimation, as previously explained. This metric provides an overview of how good is the data set, evaluates the robustness of the models with different sourced data and suggests a path for the project to take moving forward.

IV. Experimental Results

A. Data set

The available images for training, testing and validating were taken in 2018, with a total number of 174 540x960 pixel images, corresponding to the last two phases of the grape's maturation, veraison and harvest. As for the data distribution, 80% was used for training, 10% for test and 10% for validation. After dividing the images between the different sets, data augmentation and image formatting operations were performed on the training set. The aforementioned operations were different for the training and the remaining sets, since no data augmentation could be applied to the latter, tainting the results. So, for the training set, a siding window, as mentioned before, was passed trough the image, creating 6 images for each original image which turned into 12 with the inversion of each image around the vertical axis. Besides the pre processing of the training set, the other two sets could only generate 2 388x388 images per original image. With this information, the data can be summed in Table I:

For the second part of testing, the yield estimation evaluation, the 2019 data set was made available. This data set is comprised of 4 different testing conditions, 2 grape varieties, encruzado and arinto, at two stages of maturation, veraison and harvest, with a total of 40 meters per combination, resulting in 160 meters of vine over 197 540x960 pixel images.

TABLE I: Number of images of each data set before and after data augmentation and image formatting operations: training, test and validation

	Number of images(pixel area)	
Data set	Before data augmentation	After data augmentation
Training	140	1680
Test	17	34
Validation	17	34

B. Segmentation Experiments

1) Naive Baseline: As mentioned before in Chapter II, the U-Net neural network has as perks to its use the fact that it can be trained end-to-end with smaller data sets such as the one available and present satisfying results [22]. This was one of the main reasons behind its choice as the segmentation network for this work. This baseline experiment used the U-Net as it is presented in Chapter III, trained with binary cross entropy as the loss function, since that in [37] is stated that cross entropy may be more robust in maintaining its performance advantage for problems with limited data when compared to a squared-error function, no data augmentation, no pre or post processing of any kind and for 100 epochs.

This experiment resulted in all the images being 100% classified as background, with a mean of 96% accuracy and 0% IOU. From this, several hypotheses were made regarding the failure of the experiment, namely: the loss function is not taking into consideration the existing class imbalance in the images; there is not enough data for the machine to learn from; the images may need some previous processing before being used for training.

2) Loss function: As explained in Chapter III, the loss function was tweaked. Although there is a rationale behind the combination proposed previously, other weight combinations were tested. The results for the validation set during the 100 epochs of training are shown in Figure 13, which prove that best combination for the weights it is the inversion of the ratio of their imbalance, as was expected:

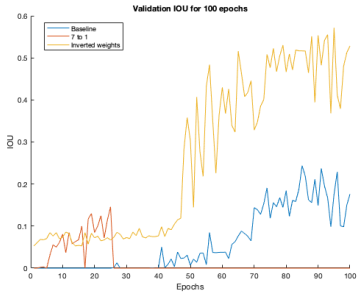


Fig. 13: Evolution of the validation IOU over 100 epochs for three different sets of weights

3) Data augmentation: As previously done in [22], to solve the problem of insufficient data, data augmentation was performed. The image was then inverted over the vertical axis. This data augmentation technique led to better values in the metrics used, especially during the first 50 to 60 epochs. Using the weighted binary cross entropy with the weight set defined

in the last section(IV-B2), there was a clear improvement (Figure 14).

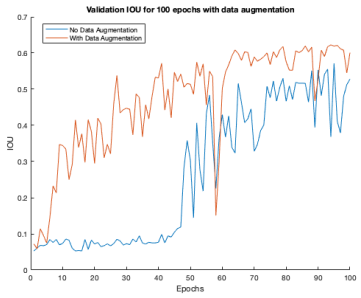


Fig. 14: Evolution of the validation IOU over 100 epochs for the set with and without data augmentation

The use of data augmentation resulted in a score that was higher than the previously best, achieving 62% validation IOU over the 57% obtained without the augmentation. Following the augmentation operations, the next step of experimentation is focused on the stages before and after the segmentation. As for the pre operations, the colour space is changed. After the segmentation, the morphological operations are tested.

4) Colour space: As stated before in Chapter III, the used colour space for this work was the CIELAB. Besides the advantages of the use of this colour space that were already described, testing was also performed for empirical confirmation. The results showed that when using this colour space, for the same test set, the results improved when compared to the RGB colour space, going from 49% IOU to 62%. It is possible to assume that this improvement comes from discarding the negative effect of illumination, as explained before in Chapter III.

5) Morphological filters: The use of morphological filters, as it is mentioned before in Chapter III, is mainly to reduce the number of false positives in the image, mostly related to small miss identified clusters. The ”open” operation was able to remove a significant part of these clusters and smoothed the grape bunch boundaries, also increasing the IOU. In the example, Figure 15, the IOU increases from 70% to 72%. Although the amount of pixels removed may be small, the difference in the metric is notorious, demonstrating the sensible nature of the problem.

(a) Original image **(b)** Resulting mask **(c)** ”Open” operation

Fig. 15: Example of the morphological operation applied on a processed image

6) Segmentation results: After all the testing, the overall process was changed according to the results. The test set produced the following scores for the described experiments (Table II):

TABLE II: IOU scores for the grape bunch class in the test set in every experiment

Experiment	Test set IOU results
Naive Baseline	0%
Loss function	47%
Data augmentation	49%
Colour space change	62%
Morphological filters	64%

C. Yield estimation experiments

With the testing made for the first stage of evaluating the system, the next step is evaluating the whole system regarding yield estimation. For this purpose another data set is available for evaluation, the 2019 set. In 2018, the project group used a different model than the one used in 2019. This model did not take into consideration the vine's porosity and as a consequence no porosity data is available. Without this data, it is not possible to use this 2018 set for yield estimation. As for the 2019 data set, it was not possible to use it for the segmentation evaluation since it does not have a complete segmentation ground truth, only the final weights. For this experiment, four situations were examined, two varieties, encruzado and arinto and two stages, veraison and harvest, the last two stages of the maturation process.

All data sets, for both arinto and encruzado, are composed of several images of four Smart Points(SP), each a set of ten consecutive meters. The SP are the same between maturation stages. It was expected that the harvest results would be more precise than the veraison, since the data was taken closer to the actual harvest, inducing less error in the models. This was proven not to be true with the veraison sets showing promising results, and the harvest sets resulting in a complete miss estimation. The possible reasons behind the results are explained in the end of this chapter.

1) Veraison: As for the veraison analysis, the results are very satisfactory. As it is possible to observe from Figure 16, the lighting conditions are similar to the training data set and constant through out the canopy. Although there are some discrepancies in the meter wise analysis, the final result shows that the overestimation compensates the underestimation, resulting in Table III, where the encruzado variety has only 3% relative error, just over 2kg over the yield and the arinto variety has 10% relative error, 13 kg less than the ground truth.

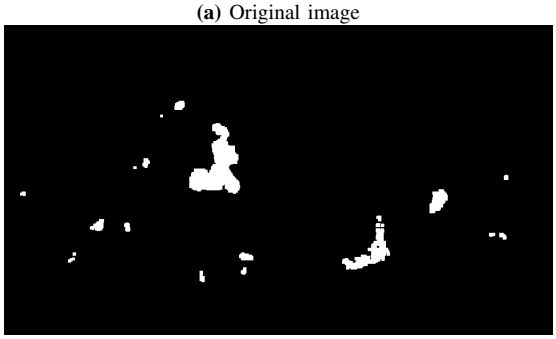
TABLE III: Final results for the yield estimation in veraison stage with the associated relative error between parenthesis

	Encruzado	Arinto
Actual yield	69,2	123,4
Hand segmentation	67,6(2%)	121,3(2%)
Automatic segmentation	71,0(3%)	110,6(10%)

2) Harvest: The summed results that make the final yield estimation are given in Table IV.

With the data from the four data sets, it is possible to understand the importance of the data collection conditions, for them to be consistent and aware of environmental factors such as lighting. For both the encruzado and arinto the results were not satisfactory, especially considering that the manual

(a) Original image



(b) Resulting mask

Fig. 16: Example of an image of the encruzado variety in veraison stage

TABLE IV: Final results for the yield estimation in harvest stage with the associated relative error between parentheses

	Encruzado	Arinto
Actual yield	69,2	123,4
Hand segmentation	68,6(1%)	122,5(1%)
Automatic segmentation	143,50(107%)	209,68(70%)

segmentation produced equally good results as before when estimating in the veraison stage. This phenomena has several possible explanations.

Firstly, these were not the expected results, especially taking into consideration the veraison estimations. The light conditions were a significant factor in the segmentation errors. Starting with the encruzado set, when compared to the veraison images, there are some noticeable differences. Although the mean light intensity in both sets is similar, the standard deviation is not for some of the images. Taking as an example the first SP in both sets, the average standard deviation of light intensity in 3 different representations, all normalised to 255, CIELAB, HSV and grey scale are of 81 points, for the harvest set and of 58 for the veraison. The difference is of 23, which is almost 10%.

This indicates that the network may not be robust enough to handle significant differences in lighting conditions, even though part of the pre processing was design to do just so. Other than the light distribution in the image, another issue is the light source and its position. Until the start of this work, the data collection was made with no specific rules regarding illumination. It was made without taking into consideration that different times of day and positions relative to the robot may interfere in the automatic segmentation. This is not the case for the veraison data, since it was taken either with the light source behind the canopy or on a cloudy day where the light is constant from any side, as is in Figure 16a. Other than the light issue, another correlation that was found was between the porosity values, the yield's ground truth and the absolute error in kg. Even though there is a general overestimation ate each SP, there are specific meters that stand out. There are two factors that most of these points share, a low porosity percentage and low yield. Since the network is prone to have false positives, any overestimation caused by the segmentation, which in turn was influenced by the data set's conditions, would be amplified by the model, overestimating the yield based on the overestimation of visible grape bunches. In general, the main conclusion that can be taken from these experiments is that the network is not robust enough to handle conditions that are significantly different from the ones it was trained on, because the training set is not diverse enough to be representative of those different conditions.

V. Future work and Conclusions

A. Future work

One of the most important stages in this process is the data collection. As it was seen previously, a lack of care can result in data that the network is not able to segment. Therefore, there are two aspects that need to be improved in this context: data collection conditions and the diversity of the training set. Firstly, the light conditions for the future data sets should match the conditions in the training set, as much as it is possible to control. Secondly, to handle discrepancies, the network should be retrained, this time with a more representative data set, possibly including the 2019 images when an appropriate ground truth is available. Also, in the next years, the network should be retrained with new data in order to understand if there is a significant improvement with time variability.

B. Conclusions

o address the main goal of this dissertation, a system described in Chapter III was proposed and presented. It was comprised of different parts, the main one for this work being the automatic segmentation. This part, isolated from the others of the project, was tested achieving satisfying results of up to 64% of IOU in the test set. This testing resulted in the adding of other components in the process, namely a pre and post processing, that were essential for the score of 64%.

After obtaining reasonable results in the segmentation evaluation, the final test is to use the models used on the hand made segmentation to estimate the yield for the same areas, the 40 meters of SP. The ideal stage to perform this estimation is in the veraison stage of the grape, around three months before the harvest. Experiments were made for both the veraison stage and the harvest stage, for two varieties, encruzado, that was used for training, and arinto. The harvest results, although not satisfactory, were useful to learn the importance of correct and consistent data collection and to expose fragilities in the segmentation network, more specifically, the inability to handle significantly different conditions to the ones present in the training data set.

The results for the veraison stage were significantly more promising, with relative errors to the actual yield in the accepted interval of 0 to 10%.

REFERENCES

[1] M. Bergerman, J. Billingsley, J. Reid, and E. van Henten, *Robotics in Agriculture and Forestry*. Cham: Springer International Publishing, 2016, pp. 1463–1492. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_56

[2] P. R. Clingeleffer, S. R. Martin, G. M. Dunn, and M. P. Krstic, *Crop development, crop estimation and crop control to secure quality and production of major wine grape varieties : a national approach : final report to Grape and Wine Research & Development Corporation*, S. Martin and G. Dunn, Eds. Adelaide, Australia: Grape and Wine Research & Development Corporation, 2001.

[3] G. Victorino, R. Braga, and C. M. Lopes, "The effect of topography on the spatial variability of grapevine vegetative and reproductive components," 2017.

[4] A. Milella, R. Marani, A. Petitti, and G. Reina, "In-field high throughput grapevine phenotyping with a consumer-grade depth camera," *Computers and Electronics in Agriculture*, vol. 156, pp. 293 – 306, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169918307580

[5] J. Fourie, J. Hsiao, and A. Werner, "Crop yield estimation using deep learning." Zenodo, Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.893710

[6] B. Keresztes, F. Abdelghafour, D. Randriamanga, J.-P. Da Costa, and C. Germain, "Real-time Fruit Detection Using Deep Neural Networks," in *14th International Conference on Precision Agriculture*, Montréal, Canada, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02518559

[7] R. Rudolph, K. Herzog, R. Töpfer, and V. Steinhage, "Efficient identification, localization and quantification of grapevine inflorescences in unprepared field images using fully convolutional networks," 2018.

[8] G. M. DUNN and S. R. MARTIN, "Yield prediction from digital image analysis: A technique with potential for vineyard assessments prior to harvest," *Australian Journal of Grape and Wine Research*, vol. 10, no. 3, pp. 196–198, 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-0238.2004.tb00022.x

[9] S. Martin, R. Dunstone, and G. Dunn, "How to forecast wine grape deliveries," 10 2003, p. 100.

[10] M. Cunha, H. Ribeiro, and I. Abreu, "Pollen-based predictive modelling of wine production: Application to an arid region," *European Journal of Agronomy*, vol. 73, pp. 42–54, 2 2016.

[11] J. M. Tarara, B. Chaves, L. A. Sanchez, and N. K. Dokoozlian, "Use of cordon wire tension for static and dynamic prediction of grapevine yield," *American Journal of Enology and Viticulture*, 2014.

[12] M. Reis, R. Morais, E. Peres, C. Pereira, O. Contente, S. Soares, A. Valente, J. Baptista, P. Ferreira, and J. Bulas Cruz, "Automatic detection of bunches of grapes in natural environment from color images," *Journal of Applied Logic*, vol. 10, no. 4, pp. 285 – 290, 2012, selected papers from the 6th International Conference on Soft Computing Models in Industrial and Environmental Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570868312000535

[13] S. Liu, S. Cossell, J. Tang, G. Dunn, and M. Whitty, "A computer vision system for early stage grape yield estimation based on shoot detection," *Computers and Electronics in Agriculture*, vol. 137, pp. 88 – 101, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916311334

[14] B. Millan, A. Aquino, M. P. Diago, and J. Tardaguila, "Image analysis-based modelling for flower number estimation in grapevine," *Journal of the Science of Food and Agriculture*, vol. 97, no. 3, pp. 784–792, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jsfa.7797

[15] S. Liu, X. Li, H. Wu, B. Xin, J. Tang, P. R. Petrie, and M. Whitty, "A robust automated flower estimation system for grape vines," *Biosystems Engineering*, vol. 172, pp. 110 – 123, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511017304610

[16] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan, and S. Singh, "Automated visual yield estimation in vineyards," *Journal of Field Robotics*, vol. 31, no. 5, pp. 837–860, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21541

[17] A. Aquino, M. P. Diago, B. Millán, and J. Tardáguila, "A new methodology for estimating the grapevine-berry number per cluster using image analysis," *Biosystems Engineering*, vol. 156, pp. 80 – 95, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511016300940

[18] N. Atif, M. Bhuyan, and S. Ahamed, "A review on semantic segmentation from a modern perspective," in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2019, pp. 1–6.

[19] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," 2017.

[20] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 87–93, 2018. [Online]. Available: https://doi.org/10.1007/s13735-017-0141-z

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[23] R. Pérez-Zavala, M. Torres-Torriti, F. A. Cheein, and G. Troni, "A pattern recognition strategy for visual grape bunch detection in vineyards," *Computers and Electronics in Agriculture*, vol. 151, no. May, pp. 136–149, 2018. [Online]. Available: https://doi.org/10.1016/j.compag.2018.05.019

[24] W. Lee, R. Ehsani, J. Schueller, V. Alchanatis, and H. Gan, "Immature green citrus fruit detection using color and thermal images," *Computers and Electronics in Agriculture*, vol. 152, no. July, pp. 117–125, 2018. [Online]. Available: https://doi.org/10.1016/j.compag.2018.07.011

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016.

[26] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] E. Chavolla, D. Zaldivar, E. Cuevas, and M. Cisneros, *Color Spaces Advantages and Disadvantages in Image Color Clustering Segmentation*, 01 2018, pp. 3–22.

[29] "Cielab colour space," https://sensing.konicaminolta.asia/what-is-cie-1976-lab-color-space/, accessed: 2020-08-13.

[30] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019. [Online]. Available: https://books.google.pt/books?id=DOamDwAAQBAJ

[31] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016.

[32] A. Saravanan, G. Perichetla, and D. K. S. Gayathri, "Facial emotion recognition using convolutional neural networks," 2019.

[33] R. Szeliski. (2011) Computer vision algorithms and applications. London; New York. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-935-0

[34] G. Victorino, G. Maia, J. Queiroz, R. Braga, J. Marques, J. Santos-Victor, and C. Lopes, "GRAPEVINE YIELD PREDICTION USING IMAGE ANALYSIS - IMPROVING THE ESTIMATION OF NON-VISIBLE BUNCHES G." Rhodes Island, Greece: 12th EFITA International Conference, 2019, pp. 60–65.

[35] R. Bonaria, "Grapevine yield estimation using image analysis for the variety arinto," Master's thesis, Instituto Superior de Agronomia, 2019.

[36] J. Queiroz, "Estimativa da produção de uva na casta Encruzado com recurso a análise de imagem," Master's thesis, Instituto Superior de Agronomia, 2018.

[37] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005. [Online]. Available: https://doi.org/10.1007/s00521-005-0467-y