

Towards Practical 2D Grapevine Bud Detection with Fully Convolutional Networks

Wenceslao Villegas Marset^{a,*}, Diego Sebastián Pérez^a, Carlos Ariel Díaz^a,
Facundo Bromberg^{a,b}

^aUniversidad Tecnológica Nacional. Dpto. de Sistemas de la Información. Grupo de Inteligencia Artificial DHARMa, Mendoza, Argentina.

^bConsejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina.

Abstract

In Viticulture, visual inspection of the plant is a necessary task for measuring relevant variables. In many cases, these visual inspections are susceptible to automation through computer vision methods. Bud detection is one such visual task, central for the measurement of important variables such as: measurement of bud sunlight exposure, autonomous pruning, bud counting, type-of-bud classification, bud geometric characterization, internode length, bud area, and bud development stage, among others. This paper presents a computer method for grapevine bud detection based on a *Fully Convolutional Networks MobileNet* architecture (**FCN-MN**). To validate its performance, this architecture was compared in the detection task with a strong method for bud detection, Scanning Windows (**SW**) based on a patch classifier, showing improvements over three aspects of detection: *segmentation*, *correspondence identification* and *localization*. The best version of FCN-MN showed a detection F1-measure of 88.6% (for true positives defined as detected components whose intersection-over-union with the true bud is above 0.5), and false positives that are small and near the true bud. Splits –false positives overlapping the true bud– showed a mean segmentation precision of 89.3%(21.7), while false alarms –false positives not overlapping the true bud– showed a mean pixel area of only 8% the area of a true bud, and a distance (between mass centers) of 1.1 true bud diameters. The

*Corresponding author

Email addresses: diego.villegas@alumnos.frm.utn.edu.ar (Wenceslao Villegas Marset), sebastian.perez@frm.utn.edu.ar (Diego Sebastián Pérez), carlos.diaz@frm.utn.edu.ar (Carlos Ariel Díaz), fbromberg@frm.utn.edu.ar (Facundo Bromberg)

paper concludes by discussing how these results for FCN-MN would produce sufficiently accurate measurements of bud variables such as *bud number*, *bud area*, and *internode length*, suggesting a good performance in a practical setup.

Keywords: Computer vision, Fully Convolutional Network, Grapevine bud detection, Precision viticulture

1. Introduction

For decades, viticulturists have been producing models of the most relevant plant processes for determining fruit quality and yield, soil profiling, or vine health, and have been gathering a wealth of information to feed into these models. Better and more efficient measuring procedures have resulted in more information, with its corresponding impact on the quality of model outcomes. Such information corresponds to a long list of variables for assessing the state of different parts of the plant, as the one found in the manual published by [The Australian Wine Research Institute \(a,b\)](#). Most of these variables of interest, however, are still being measured with manual instruments and visual inspection. This results in high labor costs that limit measurement campaigns to only small data samples which, even with the use of statistical inference or spatial interpolation techniques ([Whelan et al., 1996](#)), restrict the quality of the decisions that agronomists can conduct from them.

Precision viticulture in general ([Bramley, 2009](#)), and computer vision algorithms in particular, have been growing in the last couple of decades mostly due to their potential for mitigating these limitations ([Seng et al., 2018; Matese and Di Gennaro, 2015](#)). These algorithms come along with the promise of an unprecedented boost in the production of vineyard information as well as many expectations not only about possible improvements in the quality of the measurements, but in its potential to produce better models by feeding all this information to big data algorithms.

The present work contributes to this general endeavor with FCN-MN ¹ ([Long et al., 2015; Shelhamer et al., 2017](#)), an algorithm for measuring variables related

¹Both code and data have been made available online at <https://github.com/WencesVillegasMarset/DL4BudDetection>. The shared repository includes both the corpus of images used for training and testing, as well as runnable code for inspecting and visualizing

to one specific plant part: the bud, an organ of major importance as it is the growing point of the fruits, containing all the plant’s productive potential (May, 2000). The present contribution of autonomous bud detection not only enables the autonomous measurement of bud-related variables currently measured by agronomists (see Table 1 for a non-exhaustive list of bud-related variables), but it also has the potential to enable the measurement of novel, yet important, variables that at present cannot be measured manually. One example is the total sunlight captured by buds, which depends on the unfeasible manual task of determining the exact location of buds in 3D space. Although the present work focuses on 2D detection, it could be easily upgraded to 3D by, for instance, integrating 2D detection into the workflow proposed by Díaz et al. (2018).

Table 1 shows a non-exhaustive list of the main bud-related variables currently measured by vineyard managers (Sánchez and Dokoozlian, 2005; Noyce et al., 2016; Collins et al., 2020), together with an assessment of the extent to which detection contributes to their measurement. The right-most column (other required operations) indicates the information beyond detection, necessary to complete the measurement, while the middle columns labeled (i), (ii), and (iii) indicate the specific aspects of detection required for that variable: (i) whether it requires a good *segmentation*, i.e., the discrimination of which pixels in the scene correspond to buds and which correspond to non-bud; (ii) a good *correspondence identification*, i.e., discrimination of bud pixels as belonging to different buds; or (iii) a good *localization*, i.e., the localization of the bud within the scene. For instance, regarding the *bud number* variable, for it to coincide with the detection count, different components detected for the same bud must be bundled together as a single detection. For the *type-of-bud classification*, in addition to correctly identifying components with buds, the segmentation of the part of the image corresponding to the bud must minimize the noise produced by background pixels. Lastly, to measure the *incidence of sunlight on the bud*, localization rather than segmentation is necessary, plus the leaf 3D surface geometry.

the complete set of results of our experiments, embedding the various models of the FCN-MN detector in variable measurement systems, or re-training the FCN-MN on user provided images.

Variable	(i)	(ii)	(iii)	Other required operations
Bud number		x		none
Bud area	x	x		none
Type-of-bud classification	x	x		plant structure (trunk and canes)
Bud development stage	x	x		classifier over bud mask
Internode length (by bud detection)		x	x	plant structure (trunk and canes)
Bud volume				3D reconstruction
Bud development monitoring	x	x	x	none
Incidence of sunlight on the bud		x	x	3D reconstruction, leaves 3D surface geometry

Table 1: A non-exhaustive list of important bud-related variables accompanied by an assessment of the extent to which detection contributes to their measurement. The right-most column indicates the information beyond detection necessary to complete the measurement, while the middle columns labeled (i), (ii), and (iii) indicate the three aspects of detection required: segmentation, correspondence identification, or localization, respectively.

55 A good detector, therefore, should be evaluated on all three aspects of seg-
 56 mentation, correspondence identification and localization. This is easy for our
 57 detector as its implementation first produces a segmentation mask, which is
 58 then post-processed to produce correspondence identification and localization.
 59 The specific aspects of this approach are detailed in Section 2. The analysis of
 60 detection results presented in Section 3 shows that this approach is superior to
 61 state-of-the-art algorithms for grapevine bud detection. Finally, Section 4 dis-
 62 cusses the scope, limitations of the results obtained for bud detection, sufficiency
 63 of the performance achieved for the measurement of a selection of variables in
 64 Table 3, as well as the most important conclusions, future work and potential
 65 improvements.

66 1.1. Related work

67 A wide variety of research using computer vision and machine learning algo-
 68 rithms to acquire information about vineyards (Seng et al., 2018) can be found
 69 in the literature, such as berry and bunch detection (Nuske et al., 2011), fruit
 70 size and weight estimation (Tardaguila et al., 2012), leaf area indices and yield
 71 estimation (Diago et al., 2012), plant phenotyping (Herzog et al., 2014a,b), au-
 72 tonomous selective spraying (Berenstein et al., 2010), and more (Tardáguila
 73 et al., 2012; Whalley and Shanmuganathan, 2013). Among the outstanding
 74 computer algorithms in recent years, *artificial neural networks* have aroused

great interest in the industry as a means to carry out various visual recognition tasks (Hirano et al., 2006; Kahng et al., 2017; Tilgner et al., 2019). In particular, *Convolutional Neural Networks* (**CNN**) have become the dominant machine learning approach to visual object recognition (Ning et al., 2017). Two recent studies have successfully applied visual recognition techniques based on *deep learning networks* to identify viticultural variables to estimate production in vineyards. One of them, Grimm et al. (2019), uses an FCN to carry out segmentation of grapevine plant organs such as young shoots, pedicels, flowers or grapes. The other, Rudolph et al. (2018), uses images of grapevines under field conditions that are segmented using a CNN to detect inflorescences as regions of interest, and over these regions, the *circle Hough Transform* algorithm is applied to detect flowers.

Several works aim at detecting and locating buds in different types of crops by means of autonomous visual recognition systems. For instance, Tarry et al. (2014) presents an integrated system for chrysanthemum bud detection that can be used to automate labour intensive tasks in floriculture greenhouses. More recently, Zhao et al. (2018) presented a computer vision system used to identify the internodes and buds of stalk crops. To the best of our knowledge and research efforts, there are at least four works that specifically address the problem of bud detection in the grapevine by using autonomous visual recognition systems. The research work by Xu et al. (2014), Herzog et al. (2014b) and Pérez et al. (2017) apply different techniques to perform 2D image detection involving different computer and machine learning algorithms. In addition, Díaz et al. (2018) introduces a workflow to localize buds in 3D space. The most relevant details of each are presented below.

Xu et al. (2014)'s study presents a bud detection algorithm using indoor captured RGB images and controlled lighting and background conditions specifically to establish a groundwork for an autonomous pruning system in winter. The authors apply a threshold filter to discriminate the background of the plant skeleton, resulting in a binary image. They assume that the shape of buds resembles corners and apply the *Harris corner detector* algorithm over the binary image to detect them. This process obtains a recall of 0.702, i.e., 70.2% of the buds were detected.

108 Herzog et al. (2014b)'s work presents three methods for the detection of buds
109 in very advanced stages of development when the buds have already burst and
110 the first leaves are emerging. All methods are semi-automatic and require human
111 intervention to validate the quality of the results. The best result is obtained
112 using an RGB image with an artificial black background and corresponds to a
113 recall of 94%. The authors argue that this recall is enough to solve the problem
114 of phenotyping vines. They also argue that these good results can be explained
115 by the particular green color and the morphology of the already sprouting buds
116 of approximately 2cm.

117 Pérez et al. (2017) outlines an approach for the classification of bud images
118 in winter, using *SVM* as a classifier and *Bag of Features* to compute visual
119 descriptors. They report a recall of over 90% and an accuracy of 86% when
120 sorting images containing at least 60% of a bud and a ratio of 20-80% of bud
121 vs. non-bud pixels. They argue that this classifier can be used in algorithms for
122 2D localization of the *sliding windows* type due to its robustness to variation in
123 window size and position. It is precisely this idea that has been reproduced in
124 the present work to implement the baseline competitor to our approach.

125 Finally, Díaz et al. (2018) introduces a workflow for the localization of buds
126 in 3D space. The workflow consists of five steps. The first one reconstructs a 3D
127 point cloud corresponding to the grapevine structure from several RGB images.
128 The second step applies a 2D detection method using the sliding window and
129 patch classification technique of Pérez et al. (2017). The next step uses a voting
130 scheme to classify each point in the cloud as a bud or non-bud. The fourth step
131 applies the *DBSCAN* clustering algorithm to group points in the cloud that
132 correspond to a bud. Finally, in the fifth step, the localization is performed,
133 obtaining the center of mass coordinates of each 3D point cluster. They report
134 a recall of 45% and a precision of 100% and a localization error of approximately
135 1.5cm, or 3 bud diameters.

136 Although these research studies represent a great advance in relation to the
137 problem of detecting and localizing buds, they still show at least one of the
138 following limitations: (i) use of artificial background outdoors; (ii) controlled
139 lighting indoors; (iii) need for user interaction; (iv) bud detection in very ad-
140 vanced stages of development; (v) low bud detection/classification recall, and

141 (vi) although some of these works perform some kind of segmentation process as
142 part of the approach, none of them aim to segment the bud or report metrics of
143 the quality of the segmentation performed. These limitations represent a major
144 barrier to the effective development of tools for measuring bud-related variables.

145 2. Materials and Methods

146 2.1. Fully Convolutional Network with MobileNet (FCN-MN)

147 As outlined in the introduction, the approach proposes the use of computer
148 vision algorithms to: (i) *segment* buds by *classifying* which pixels in the scene
149 correspond to buds and which correspond to background (non-buds), (ii) *identify*
150 bud *correspondences* by discriminating those pixels that belong to different buds
151 in the observed scene, and (iii) *localize* each bud in the scene.

152 For the segmentation operation, i.e., pixel classification, the fully convolutional
153 network introduced in ([Long et al., 2015](#)) is taken as a basis and trained
154 for the specific problem of grapevine bud segmentation. The following section
155 [2.1.1](#) describes in detail the architecture considered for these networks. The re-
156 sulting fully convolutional network returns a probability map on the same scale
157 as the original image, where the value of one pixel represents the probability
158 that the corresponding pixel in the input image belongs to a bud. To obtain a
159 binary mask, a binarization threshold τ with values $\{0.1, 0.2, \dots, 0.9\}$ is applied
160 to each pixel, classifying the pixel as bud (non-bud) if its probability is higher
161 (lower) than τ . To identify bud correspondences, post-processing of this binary
162 mask is performed to determine that two bud pixels correspond to the same bud,
163 as long as they belong to the same connected component, i.e., joined by some
164 sequence of contiguous bud pixels. Finally, there are several alternatives for the
165 localization of objects among which are *bounding box*, *pixel-wise segmentation*,
166 *contour* and *center of mass* of the *object* ([Lampert et al., 2008](#)). In this work
167 the last one was considered, choosing to localize buds by the center of mass of
168 the connected component.

169 2.1.1. Encoder-decoder architecture

170 For the pixel classifier, the three versions –32s, 16s and 8s– of the *fully con-*
171 *volutional networks* originally introduced by ([Long et al. \(2015\)](#)) were considered,

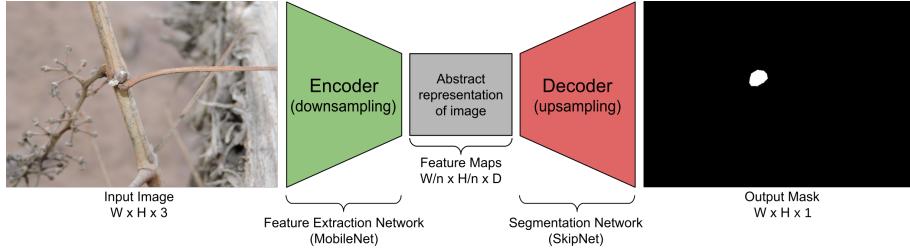


Figure 1: Diagram of the FCN-MN network architecture proposed in this work, based on the fully convolutional network proposed by [Shelhamer et al. \(2017\)](#), replacing its feature extraction encoder with the MobileNet network [Howard et al. \(2017\)](#), which produces feature maps with a downsampling factor of n . As a decoder for the production of the segmentation map, the SkipNet network [Siam et al. \(2018\)](#) is used, implementing variants 32s, 16s and 8s.

mainly due to their promising results in many image segmentation applications ([Litjens et al., 2017](#); [Garcia-Garcia et al., 2018](#); [Kaymak and Uçar, 2019](#)). These networks have characteristic architectures with two distinct parts: *encoder* and *decoder* (see Figure 1).

The encoder consists of a convolutional neural network that performs a *downsampling* of an input image into a feature set, by means of convolution operations to produce a set of *feature maps*, i.e., an abstract representation of the image that captures semantic and contextual information, but discards fine-grained spatial information. These operations reduce the spatial dimensions of the image as one goes deeper into the network, resulting in feature maps $1/n$ the size of the input image, where n is the downsampling factor. The decoder is an *upsampling* subnet, which takes the low-resolution feature map and projects it back into pixel space, increasing the resolution to produce a segmentation mask (or dense pixel classification) with the same dimensions as the input image. This operation is implemented as a network of transposed convolutions with trainable parameters, also known as upsampling convolutions ([Shelhamer et al., 2017](#)).

To refine the segmentation quality, connections that go beyond at least one layer of the network, called *skip connections*, are often used to transfer local spatial information from the internal encoder layers directly to the decoder. In general, these connections improve segmentation results, since they mitigate the loss of spatial information by allowing the decoder to incorporate information

from internal feature maps. Their impact may vary depending on the proposed skip architecture. In Long et al. (2015), three skip architectures are proposed: 32s without information from internal encoder layers; 16s that adds spatial information from deep encoder layers; and 8s that adds spatial information from deep and less deep encoder layers. The details of these architectures are beyond the scope of this paper, but can be found in Long et al. (2015) and Shelhamer et al. (2017). Since the results reported in the literature are not conclusive regarding which architecture is better, in this work all three alternatives are considered.

In spite of having achieved excellent results in practice, these architectures carry a significant load of computational resources. With this in mind, in this work the VGG encoder of Simonyan and Zisserman (2015), originally proposed by Long for fully convolutional networks, was replaced by the MobileNet network of Howard et al. (2017), thus the suffix MN in the name of the FCN-MN algorithm. This network stands out for having only 4.2 million parameters against the 138 million parameters of VGG, allowing the training and testing process to be considerably faster, with a much lower memory requirement. This situation was verified by preliminary experimentation, in which indeed MobileNet ended as the fastest, less memory intensive option for our training specification and hardware available. This experimentation is outside the scope of this manuscript and thus further details have been omitted. The use of MobileNet as an encoder in the fully convolutional networks of Long et al. (2015) is not new, but had already been proposed for the 8s architecture by Siam et al. (2018) in his SkipNet architecture. Technically, Siam et al. (2018)'s proposal is extremely simple; motivating us to extend it to the 16s and 32s architectures originally proposed by (Long et al., 2015).

220 2.2. Sliding Windows detector

221 This section describes both, the approach proposed by Pérez et al. (2017)
222 for the classification of bud images, and our implementation for detection based
223 on the sliding windows outlined in the original paper, denoted hereon by **SW**.
224 Details of the six steps of the proposed SW detection procedure are shown in
225 Figure 2.



Figure 2: Diagram of the SW bud detection approach based on Pérez et al. (2017). Multiple patches from the input image shown in (a) are extracted via the Sliding Windows algorithm (b). Then, in step (c) a SIFT + BoF descriptor is computed for each patch. These descriptors are classified by an SVM binary classifier (d) in order to determine whether a bud is present on each patch. Finally, the generation of a binary segmentation mask is achieved in two steps. First, in step (e) a voting scheme is applied to each pixel, assigning it one vote for each positive patch it belongs to. Then, the pixel is incorporated in the output segmentation mask if the number of votes it obtained is more or equal to a user given threshold ν . Finally, step (f) shows the output mask corresponding to $\nu = 3$ (black) together the ground truth segmentation (white).

226 In the present work, different variations of the SW algorithm are contem-
227 plated, considering squared windows of the 10 sizes 100, 200, 300, 400, 500,
228 600, 700, 800, 900 and 1000 pixels, and the four values {1, 2, 3, 4} for the voting
229 threshold ν . These window sizes were chosen on the basis of the robustness
230 analysis of the classifier presented by Pérez et al. (2017) for the window geom-
231 etry. This analysis shows that the classifier is robust for patches that contain
232 at least 60% of the pixels of a bud, and whose area is composed of at least
233 20% bud pixels. If we consider extreme cases, i.e., the smallest bud diameter
234 of 100px and the largest of 1600px, window sizes of 100px and 1000px could
235 contain at least 60% of the pixels of a bud. In addition, using a 50% displace-
236 ment, it is guaranteed that at least one patch will contain more than 20% bud
237 pixels, 50px and 500px, respectively. The authors argue that a sliding window
238 detection algorithm could easily propose a scheme for choosing window size and
239 displacement to ensure that at some point in the scan the window meets the
240 robustness requirements. However, no details are given on how to implement it,
241 so in this paper we only report results for fixed window sizes and 50% vertical
242 and horizontal displacement. As a result, each pixel of the image simultaneously
243 belongs to 4 patches, which justifies the maximum threshold value of 4.

244 For all remaining parameters we considered a single value. The parameter
245 values chosen for the Bag of Features and SVM algorithms (step c) are those of
246 the original publication of Pérez et al. (2017), and are discussed in Section 2.3.3
247 together with details of the algorithms' training.

248 2.3. Model training

249 This section provides details of the training process for each approach. In
250 order to contrast both approaches they have been designed to receive the same
251 type of input, i.e., an image of a viticultural scene, and to produce the same
252 outputs, i.e., a binary mask of the same size as the original image whose positive
253 pixels represent bud-type pixels. This allows both algorithms to be trained with
254 the same image collection, which is described in the following section, followed
255 by model-specific training details.

256 2.3.1. *Image collection*

257 The image collection used in this study is the same collection originally used
258 in Pérez et al. (2017), which has been downloaded from [http://dharma.frm.
259 utn.edu.ar/vise/bc](http://dharma.frm.utn.edu.ar/vise/bc) as indicated by the authors. The collection corresponds
260 to bud images captured in winter in natural field conditions, on approximately
261 a hundred *Vitis Vinifera* plants from 10 different varieties, all driven by a trellis
262 system. The complete collection consists of 760 images. However, in this work,
263 only images containing exactly one bud were kept from the original dataset, re-
264 sulting in a corpus of 698 images. Cases with more than one bud were scarce for
265 a proper training of the FCN-MN architecture. This may restrict the practical
266 applications of the trained detection model by forcing them to use frames with
267 only one bud. However, training and evaluating a one-bud model lays a strong
268 groundwork for any future work, both academic or technological, that could
269 overcome this limitation by producing the necessary multi-bud training corpus.
270 Each image in the corpus is accompanied by the ground truth, that is, a mask of
271 the manual segmentation of the bud. These images and their masks were used
272 during the training and evaluation of the detection models. For this purpose,
273 the image collection was separated into two disjoint subsets: the *train set* with
274 80% of the images and the *test set* with the remaining 20%. This resulted in a
275 train set of 558 images and a test set of 140 images, both with their respective
276 ground truth masks.

277 2.3.2. *FCN-MN training*

278 The 558 images reserved for this purpose were used to train this approach.
279 These images have different resolutions; however, the three proposed FCN-MNs
280 require a fixed size entry. Therefore, all images (including their masks) were
281 scaled to a resolution of 1024×1024 pixels using a bilinear interpolation method
282 (Han, 2013). In addition, for the train set images, the pixel RGB intensity values
283 were scaled from [0; 255] to [-1; 1].

284 Given the small number of images in the train set, two techniques widely used
285 in practice were employed to achieve robust training: *transfer learning* (Pan and
286 Yang, 2009) and *data augmentation* (Shorten and Khoshgoftaar, 2019). The
287 transfer learning process was carried out as follows: (i) the original MobileNet
288 network proposed by Howard et al. (2017) was implemented; (ii) the network was

289 initialized with the parameters pre-trained on the ImageNet benchmark dataset
290 (Kornblith et al., 2019); (iii) the MobileNet multi-class classification layer was
291 replaced by a binary classification layer; (iv) the network was trained as a bud
292 and non-bud patch classifier in an analogous way to SVM training using the
293 same balanced patch train set used for training SW, after scaling all its images
294 to 224×224 pixels; and (v) the parameters obtained in the previous step were
295 used to initialize the encoder of our FCN-MN. The data augmentation process
296 was applied on the fly during training, meaning that at each iteration the trainer
297 receives one transformed version of the original image obtained by applying the
298 following seven operations to the original image over parameter values chosen
299 at random with uniform probability: *rotation* of up to 45° ; *horizontal shifting*
300 of up to 40%; *vertical shifting* of up to 40%; *shear* of up to 10%; *Zoom* of up
301 to 30%; *horizontal flip* and *vertical flip*. Given that there are 200 epochs, the
302 trainer is presented with 200 transformed versions of each image in the corpus,
303 equivalent to one large dataset of 111600 images.

304 For the training of the three FCN-MN variants –8s, 16s, and 32s– it is
305 required to specify the *optimization method* and *dropout* value, two parameters
306 typically defined by the user. In this work, the optimization methods considered
307 were: *Adam* with learning rate 0.001, *beta1* = 0.9 and *beta2* = 0.999; *RMSProp*
308 with learning rate 0.001 and ρ = 0.9; and *Stochastic Gradient Descent* with
309 learning rate 0.0001 and *momentum* = 0.9. For the dropout case, two values
310 were considered: 0.5 and 0.001. These values were pre-selected by preliminary
311 experiments not discussed here.

312 The best combination of optimization method and dropout was determined
313 in training time over a validation set, using the *4-fold cross validation* approach
314 by 60 epochs and batchsize equal to 4, varying over the three optimization
315 methods and the two dropout values. The values selected were those that max-
316 imize the mean of Jaccard’s *Intersection-over-Union* (IoU) (Jaccard, 1912), a
317 typical assessment measure in segmentation problems. For each combination of
318 optimizer and dropout values the simple mean is reported over 12 *IoUs* corre-
319 sponding to the 3 variants considered in each of the 4 folds. It can be observed
320 in Table 2 that the combination of parameters with which the highest average
321 *IoU* is reached is RMSProp with a dropout of 0.001. Using these parameters,

Optimizer	Mean IoU	
	Dropout = 0.001	Dropout = 0.5
RMSprop	<u>0.44253</u>	0.3117
Adam	0.240277	0.315714
SGD	0.000886	0.00151

Table 2: For each combination of optimizer and dropout values the simple mean is reported between 12 *IoUs* corresponding to the 3 variants considered in each of the 4 folds.

322 the 8s, 16s, and 32s architectures were trained over 200 epochs and batch size
 323 of 4.

324 *2.3.3. SW training*

325 The training of SW is conducted in the same way as for the original workflow
 326 proposed in Pérez et al. (2017). This involves training a binary classifier to learn
 327 the concept of bud versus non-bud from a collection of rectangular patches that
 328 may or may not contain a bud. During the training, bud patches must be regions
 329 that perfectly circumscribe the bud while non-bud patches must be regions that
 330 contain not a single bud pixel (see Figure 3). Therefore, to build the patch
 331 collection, the 558 images and their masks were processed following the same
 332 protocol as in Pérez et al. (2017), obtaining a total of 558 patches circumscribing
 333 each bud (one per image), and more than 25000 non-bud patches (the non-bud
 334 area is much larger than the area occupied by a bud in the image). The size of
 335 these patches is variable, with resolutions between 0.1 and 2.6 megapixels for
 336 the 100×100 to 1600×1600 pixels patches.

337 From this collection of patches, a balanced patch train set was created, with
 338 558 patches for each class, where non-bud patches were taken at random from
 339 the collection of 25000 background patches. The training was performed as
 340 detailed in the pipeline proposed by Pérez et al. (2017): (i) all SIFT descriptors
 341 were extracted from the train set; (ii) BoF was applied with a vocabulary size
 342 equal to 25; and (iii) the SVM classifier was trained on the BoF descriptors of
 343 each patch using a *Radial Basis Function* kernel, where the value of the γ and
 344 C parameters was established by means of a 5-fold cross-validation on the same
 345 value ranges: $\gamma = \{2^{-14}, 2^{-13}, \dots, 2^{-7}\}$ and $C = \{2^5, 2^6, \dots, 2^{14}\}$.



Figure 3: A sample of the collection of patches used in this work. The first and second rows correspond to bud patches and non-bud patches, respectively. Image extracted from Pérez et al. (2017).

3. Experimental results

In this section we present a systematic evaluation of the quality of our proposed FCN-MN procedure for bud detection over all three aspects of detection required for the measurement of the relevant bud-related variables listed in Table 1: *segmentation*, *correspondence identification*, and *localization*. First, in the following subsection, we present metrics that quantify the quality of these aspects, followed by subsection 3 that presents the results for the metric values obtained for different experiments over the image test set.

3.1. Performance metrics

3.1.1. Correspondence identification metrics

Detection of buds is the result of two steps: (i) thresholding of the output masks into a *binary mask*, and (ii) considering each *connected component* of the binary mask as exactly one detected bud. For FCN-MN, the thresholding is done by keeping all pixels of the probabilistic mask with values higher than τ , and for SW this is done through the voting mechanism that keeps all pixels that belong to at least ν positive patches. The correspondence identification metrics measure to what extent these detections are *correct* or *incorrect*. Detected components are considered to be correct when most of its mask coincides with the mask of the true bud. This condition is formalized by considering *true positives* as those whose *intersection-over-union* between their masks and the masks of true buds surpasses some threshold α . Denoted *IoU*, this coefficient

367 is defined as the area of the intersection between the detected and true masks,
 368 normalized by the area of their union. The *IoU* coefficient has also appeared
 369 in the literature as the Jaccard's coefficient ([Jaccard, 1912](#)), an alternative to
 370 the harmonic mean (a.k.a. F1-measure) of the pixel-wise precision and recall
 371 between the detected components and the true mask. The *IoU* coefficient runs
 372 from 0 when the detection missed completely the true bud, to 1 when the masks
 373 coincide perfectly. Values in between correspond to some true buds missing in
 374 the detected mask, or non-buds showing in the detected mask. This definition
 375 of detection may result in confusion when some detected component correctly
 376 detects more than one bud, i.e., some detected component overlaps with an *IoU*
 377 higher than 0.5 with more than one true bud. This, however, cannot occur in
 378 our experiments because the image collection contains only one bud per image.

379 This results in the following metrics for correspondence identification, de-
 380 fined for an arbitrary value of the threshold α :

- 381 • **True Positive** ($TP(\alpha)$): number of detected components with $IoU \geq \alpha$.
- 382 • **False Positives** ($FP(\alpha)$): number of detected components with $IoU < \alpha$.
- 383 • **False Negatives** ($FN(\alpha)$): number of true buds for which there is no
 384 true positive detected component, that is, no detected component with
 385 $IoU \geq \alpha$.

386 Rather than reporting these quantities individually, we combine them in the
 387 well known precision and recall metrics, denoted as $P_D(\alpha)$ and $R_D(\alpha)$, referred
 388 to as *detection-precision* and *detection-recall*, and defined formally as

$$P_D(\alpha) = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} = \frac{TP(\alpha)}{TP(\alpha) + FP(\alpha)},$$

$$R_D(\alpha) = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} = \frac{TP(\alpha)}{TP(\alpha) + FN(\alpha)}.$$

389 Given these quantities, we also report the *F1-measure*, denoted $F1(\alpha)$, com-
 390 puted as their harmonic average $F1(\alpha) = 2 \times \frac{P_D(\alpha) \times R_D(\alpha)}{P_D(\alpha) + R_D(\alpha)}$.

391 These correspondence identification metrics provide a strong summarization
392 of the merit of the detection, but may lack some refinements necessary for
393 assessing aspects of the bud detection with impact on the different possible
394 applications of bud related variable measurements. For that, it is paramount
395 to understand the impact of the false positive components. To start then, we
396 distinguish them between those that overlap a true bud on at least a single
397 pixel, named hereon as *splits*, and those that do not overlap a true bud, named
398 hereon as *false alarms*. Formally,:;

- 399 • **Split ($S(\alpha)$)**: number of detected components satisfying $0 < IoU < \alpha$,
400 i.e., are false positives but overlap the true bud.
- 401 • **False Alarm ($FA(\alpha)$)**: number of detected components with $IoU = 0$,
402 i.e., are false positives and do not overlap a true bud.

403 In the following sections, α is considered to be 0.5, the most common choice
404 in the literature of detection algorithms, with some minor exceptions considered
405 for a detailed and thorough analysis. To simplify the notation we drop (α) for
406 the cases corresponding to $\alpha = 0.5$. For instance, we replace $P_D(0.5)$, $R_D(0.5)$
407 and $F1(0.5)$ with P_D , R_D and $F1$.

408 3.1.2. Segmentation metrics

409 All correspondence identification metrics, including S and FA , are based
410 on the rather coarse binary assessment of correct or incorrect. This allows a
411 simple and summarized evaluation but may miss some subtle, pixel-wise errors,
412 and their resulting impact on the measurement of bud related variables. A true
413 positive, for instance, could miss that some component may have an IoU much
414 larger than 0.5, even 1.0, meaning that its mask is matching perfectly that of
415 the true bud. For other non perfect cases, the same IoU can be obtained for
416 many combinations of intersections and unions, with extreme cases of a small
417 detected component completely contained within the true bud presenting the
418 same IoU of a large detected component containing completely the true bud.

419 A pixel-wise comparison of the masks could also help to assess the quality
420 of false positive detections. For the case of splits, the best case would be one
421 completely enclosed within the true mask, –i.e., presenting not a single false

422 positive pixel-, while covering half minus one of the pixels of the true bud
423 mask. For the case of false alarms, a correspondence identification metric may
424 miss how large or small are these false positives.

425 The community has proposed several metrics to quantify segmentation er-
426 rors. The most obvious ones are those that report the *fraction* of the detected
427 mask corresponding to *true positive*, *false positive*, and *false negative* pixels;
428 denoted *TPF*, *FPF*, and *FNF*, respectively. Again, one can simplify the anal-
429 ysis by considering pixel-wise precision and recall, denoted as P_S and R_S and
430 referred to as *segmentation precision*, *segmentation recall*, defined formally as:

$$P_S = TPF / (TPF + FPF),$$

$$R_S = TPF / (TPF + FNF),$$

431 which can be combined by their weighted harmonic mean, the well-known
432 *F1-measure* or Dice coefficient. The *IoU* coefficient is, however, a more natural
433 choice, both for its similarity with the Dice coefficient, and the fact that it was
434 used in the definition of the correspondence identification metrics.

435 One could further refine these metrics by applying them, not to the whole
436 mask, but to the individual correspondence identification cases; for instance,
437 by reporting the mean *IoU* over only true positive components. Also, one
438 could apply them only to splits to assess how bad or good splits are, meaning,
439 how much extra area of the true bud is detected by them. The case of false
440 alarm detections is rather monotonous and not very informative as its precision
441 and recall is always zero. Instead, these components can be better assessed
442 by considering a normalization against bud size to measure their relative size,
443 resulting in the *normalized area*, denoted as *NA* and defined formally as *the*
444 *area of the component normalized by the area of the (single) true bud in the*
445 *image*, with a component's area corresponding to its total number of pixels.

446 3.1.3. Localization metrics

447 As a complement to the segmentation metrics we consider the localization
448 of the detected components. Mostly useful for false alarms by noticing that
449 false alarms at different distances of the true bud may affect differently the

450 measurement of some bud related variables. In some cases, with proper post-
451 processing (e.g. spatial clustering), the impact of near-by false alarms on the
452 overall error in the measurement of bud related variables may be reduced or
453 could even disappear.

454 The selected metric for assessing the localization error of detected compo-
455 nents is *normalized distance*, denoted as ND and formally defined as *the dis-*
456 *tance between the center of mass of the component and the center of mass of*
457 *the true bud, divided by the diameter of the true bud*, with the bud's diameter
458 corresponding to the maximum distance between any two bud pixels.

459 *3.2. Results*

460 This section validates that FCN-MN is a better detector than its SW coun-
461 terpart through a systematic assessment over each of the metrics defined in the
462 previous section..

463 For a thorough comparison, several cases for each algorithm were considered:
464 training 27 FCN-MN detectors and 40 SW detectors over the training set of 558
465 images, one for each combination of their respective hyper-parameters. For
466 FCN-MN, these hyper-parameters are the three architectures –8s, 16s, and 32s–
467 and the 9 values $\{0.1, 0.2, \dots, 0.9\}$ for the binarization threshold τ . For SW,
468 in turn, these hyper-parameters are the 10 patch sizes $\{100, 200, \dots, 1000\}$ and
469 the 4 values $\{1, 2, 3, 4\}$ of the voting threshold ν . Once trained, each of these
470 67 models were evaluated over the 140 images reserved for testing purposes,
471 obtaining for each image the detection components.

472 Table 3 shows the results for the best detectors of each algorithm, reporting
473 all performance metrics of the three aspects of detection over all detected com-
474 ponents over the 140 test images: correspondence identification, segmentation
475 and localization. The first column shows the label of the selected detectors, with
476 the subscript indicating the architecture and patch size for the case of FCN-MN
477 and SW, respectively; and the superscript indicating the thresholds τ and ν ,
478 respectively.

479 The table includes all metrics defined in Section 3.1 required for a thor-
480ough comparison of FCN-MN against SW. First, four correspondence identifi-
481cation metrics are included: detection-precision P_D , detection-recall R_D , the
482 F1-measure $F1$, and S , the total count split components, all corresponding to

483 an α of 0.5. Also, seven segmentation metrics are included: the mean and
484 standard deviation (in parenthesis) of the segmentation precision, segmentation
485 recall, and the IoU measure over the $\alpha = 0.5$ true positives and splits, denoted
486 in the table by P_S^{TP} , R_S^{TP} and IoU^{TP} and P_S^S , R_S^S and IoU^S for true positives
487 and splits, respectively; plus the mean and standard deviation of the normalized
488 area for $\alpha = 0.5$ false alarms, titled NA . Finally, the table reports the normal-
489 ized distance ND of the $\alpha = 0.5$ false alarm components, and omits the ND
490 for true positives and splits, assumed too close to the true bud to produce any
491 results of interest. This is confirmed below when their minimum and maximum
492 NDs are reported and discussed.

493 The table is a summary, as it includes only a subset of all 27 FCN-MN
494 cases and all 40 SW cases. A detector was considered for inclusion in the table
495 if, when compared to its counterparts of the same algorithm, it resulted in
496 the highest value for at least one of the metrics. The corresponding cell was
497 marked in bold in the table. For instance, the detector $FCN\text{-}MN_{16s}^{0.6}$ has been
498 included because its detection-precision P_D of 88.6% is the largest among the
499 detection-precision of all 27 FCN-MN detectors. Similarly, the detector SW_{700}^3
500 has been included because its precision $P_D = 2.5\%$ is the largest among all 40
501 SW detectors. Also, for all metrics, the best among the FCN-MN detectors
502 (bolded) has been compared to the best among the SW detectors (bolded), and
503 the larger of the two has been underlined. The table shows an overwhelming
504 improvement of FCN-MN over SW. A first analysis of the table shows FCN-MN
505 with larger metrics over SW in all cases, except for the segmentation recall R_S^S
506 for splits, for which the SW case has a better (larger) mean of 98.1% compared
507 to the 58.6% for FCN-MN. These improvements are not statistically significant,
508 however, as the large standard deviations of 50.5 for the FCN-MN cases results
509 in (statistically) overlapping values.

510 For the case of correspondence identification metrics P_D , R_D , $F1$ and S ,
511 FCN-MN values are overwhelmingly better to those of SW, with the best preci-
512 sions and recalls of SW all below 6.4% against those of FCN-MN whose values
513 surpass 30.1%. For the case of splits one can observe the same pattern, with
514 SW showing the best case of 82 splits, much larger than the 9 splits of the best
515 case of FCN-MN. Although not quite overwhelming, the segmentation metrics

of FCN-MN are still larger than those of SW. For instance, for the segmentation precision of true positives P_S^{TP} , and split P_S^S , the FCN-MN over SW improvements are 98.2% versus 94.1%, and 98.8% versus 54.2%, respectively. Finally, for NA and ND (of false alarms), where a smaller value is better, again FCN-MN shows large improvements over SW, with the best values of NA are 0.04 versus 0.23, and the best values of ND are 1.10 versus 5.97, for FCN-MN versus SW, respectively.

FCN-MN also shows improvements over the mean normalized distances of the true positives and splits. These have been computed but omitted in the table. For FCN-MN the *minimum* and *maximum* mean and standard deviations are 0.038(0.037) and 0.055(0.053), respectively. Similarly, the FCN-MN minimal and maximal pair for the split components are 0.216(0.138) and 0.482(0.212), respectively. As predicted, all rather small, with both the minimum and maximum mean distance falling well within one diameter of a true bud, for all cases. For the SW detectors, the min/max pair of mean normalized distances for the true positive components is 0.045(0.023)/0.210(0.076), and for splits components is 0.412(0.210))/3.250(5.961), respectively. As can be observed, again FCN-MN shows an improvement over SW, with a minor statistically significant overlap of their min/max intervals for both the true positives and split cases.

3.2.1. Detailed analysis of correspondence identification metrics

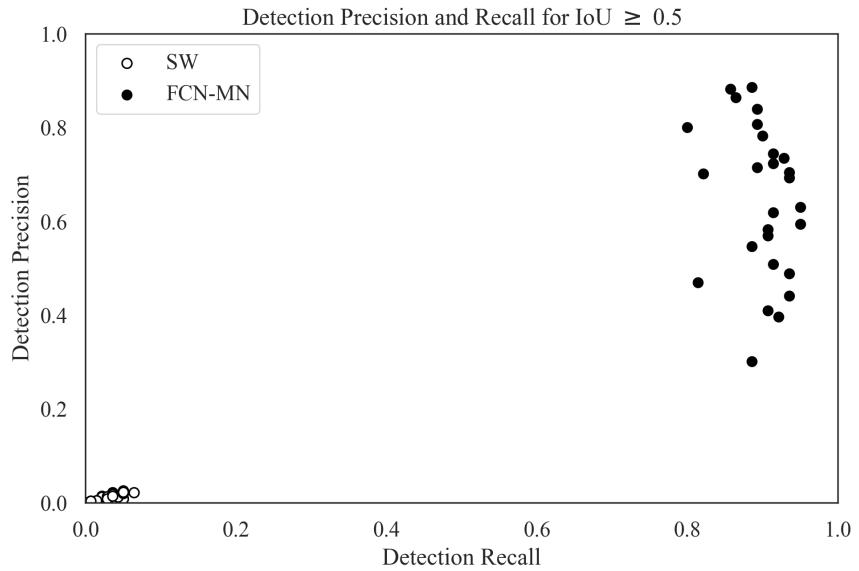
Graphically, one could expect a better combined analysis of detection-precision and detection-recall than could be obtained by comparing the F1-measure. This is shown as a scatter plot in Figure 4a, a graphical representation of a non-summarized version of the second and third columns of Table 3. Each dot in the plot is located according to the detection-precision and detection-recall, and the color black or white, whether it corresponds to an FCN-MN or an SW detection model.

The graph reinforces the clear and undisputed improvements of FCN-MN over SW already shown in the table, with overwhelmingly larger detection precisions and recalls.

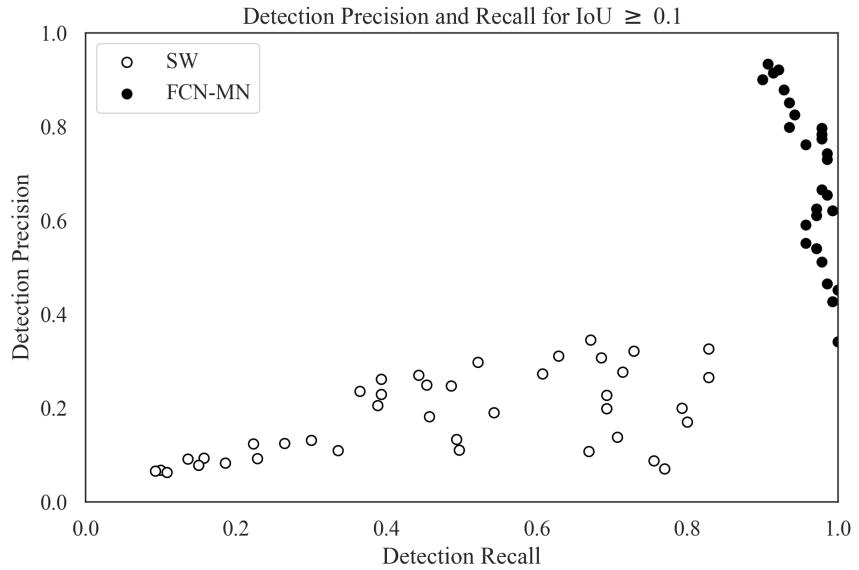
One concern that may arise is the confidence one can ascribe to such overwhelmingly bad results of the SW correspondence identification metrics. One possible explanation may arise by noticing the large number of splits of SW.

Detector	P_D	R_D	F_1	S	P_S^{TP}	R_S^{TP}	IoU^{TP}	P_S^S	R_S^S	$IoUS$	NA	ND
$FCN_{8s}^{0,1}$	39.6	92.1	55.4	17	78.8(9.1)	97.6(6.8)	76.9(8.9)	63.0(32.3)	5.8.6(50.5)	22.3(21.1)	0.17(0.81)	7.61(7.54)
$FCN_{8s}^{0,2}$	59.4	95.0	73.1	13	83.5(9.5)	96.7(6.2)	80.8(9.3)	70.5(36.7)	36.1(44.7)	15.2(17.3)	0.29(0.88)	4.73(5.45)
$FCN_{8s}^{0,3}$	63.0	95.0	75.8	15	87.4(8.5)	95.0(7.8)	83.1(9.1)	75.2(36.1)	28.4(42.7)	11.8(16.9)	0.28(0.77)	3.98(4.74)
$FCN_{8s}^{0,4}$	69.3	93.6	79.6	9	90.1(7.9)	93.8(6.9)	84.7(8.2)	71.1(32.4)	54.2(38.7)	29.5(17.7)	0.29(0.76)	3.54(4.47)
$FCN_{8s}^{0,9}$	70.1	82.1	75.7	34	98.2(5.1)	75.1(11.3)	73.8(10.6)	98.8(7.2)	17.7(19.7)	17.1(18.4)	0.24(0.5)	3.80(5.66)
$FCN_{16s}^{0,4}$	80.6	89.3	84.7	10	88.6(8.2)	93.3(8.5)	82.8(8.7)	78.6(33.6)	32.0(33.5)	22.0(16.2)	0.04(0.09)	3.80(5.08)
$FCN_{16s}^{0,6}$	88.6	88.6	88.6	10	92.8(6.7)	89.3(10.2)	83.1(9.4)	89.3(21.7)	26.9(34.1)	18.6(19.5)	0.08(0.11)	1.10(0.65)
$FCN_{32s}^{0,1}$	30.1	88.6	44.9	32	71.5(10.1)	98.2(5.5)	70.2(9.1)	69.1(30.2)	46.1(48.1)	19.2(19.5)	0.14(0.66)	4.62(5.59)
$SW_{200}^{4,0}$	2.1	6.4	3.2	142	72.2(6.0)	75.1(8.3)	58.1(6.4)	44.5(31.9)	40.1(33.8)	17.6(14.0)	1.00(1.78)	8.68(6.58)
$SW_{100}^{4,0}$	0.3	1.4	0.5	196	59.5(4.6)	85.5(16.7)	53.4(2.9)	54.2(34.8)	17.1(22.3)	11.0(12.4)	0.23(0.59)	5.97(6.51)
$SW_{1000}^{4,0}$	1.4	2.1	1.7	82	65.6(5.1)	74.0(6.0)	53.1(3.1)	20.4(17.3)	67.0(32.1)	16.3(12.4)	13.87(21.8)	7.15(5.2)
$SW_{700}^{3,0}$	2.5	5.0	3.4	109	64.0(6.0)	85.1(7.7)	57.2(4.8)	15.8(14.0)	82.1(26.1)	13.6(9.1)	15.95(28.85)	8.10(4.79)
$SW_{600}^{2,0}$	0.3	0.7	0.4	135	54.3(–)	97.1(–)	53.4(–)	10.2(10.0)	91.6(21.0)	9.8(9.5)	20.63(38.89)	7.94(4.39)
$SW_{500}^{1,0}$	0.0	0.0	0.0	140	0.0(–)	0.0(–)	0.0(–)	8.4(9.6)	98.1(9.6)	8.3(9.5)	17.39(30.06)	7.22(4.04)
$SW_{500}^{4,0}$	0.4	0.7	0.5	119	94.1(–)	70.1(–)	67.2(–)	27.9(22.3)	60.2(31.1)	19.7(12.0)	5.90(8.43)	9.53(5.76)

Table 3: Correspondence identification, segmentation and localization metrics for the best FCN-MN and SW detection models. Each column shows bolded cells corresponding to the cell with the best metric among all FCN-MN rows and the cell with the best metric among SW rows, and underlined cells corresponding to the best among all combined models, i.e., the best of the column. Columns P_D , R_D , F_1 and S show results for the *Correspondence identification metrics* detection-precision, detection-recall, F1-measure and number of images with splits, respectively; Columns P_S^{TP} , R_S^{TP} and IoU^{TP} (resp. P_S^S , R_S^S and $IoUS$) correspond to the *segmentation metrics* mean segmentation precision, mean segmentation recall, and mean IoU measure over all true positive components (resp. split components), with standard deviations in parenthesis (undefined cases denoted by “–”); and Columns NA and ND show the mean NA and mean ND over all false alarm components.



(a)



(b)

Figure 4: Scatterplots of detection-precision $P_D(\alpha)$ versus detection-recall $R_D(\alpha)$, with results for $\alpha = 0.5$ and $\alpha = 0.1$ shown in Figure (a) and (b), respectively. Results for FCN-MN and SW are shown in black and white dots, respectively. Each dot represents the detection-precision P_D and detection-recall R_D for some particular configurations of hyper-parameters among all models (27 for FCN-MN and 40 for SW).

549 Splits are components that could not pass the $IoU \geq 0.5$ condition but are
550 overlapping the true bud. This suggests that SW may be producing too many
551 small detections of the bud, all of which could not make the cut for $\alpha = 0.5$.
552 This can be confirmed by observing in Table 3 the mean IoU of splits for the
553 SW models, all of which are well below 50%, with the maximum at 19.7%.
554 We complement this by also considering correspondence identification metrics
555 with a smaller α of 0.1, whose precision-recall scatterplot is shown in Figure 4b.
556 With similar results for FCN-MN, the graph shows clear improvements for SW,
557 with precisions reaching almost 40% and recalls above 80%. The increase in
558 the detection-precision proves that many of the $\alpha = 0.5$ splits are true positives
559 for $\alpha = 0.1$. This may even result in some true buds with not a single true
560 positive for the case of $\alpha = 0.5$, may now have one, which explains the increase
561 in detection-recall.

562 3.2.2. Detailed analysis of segmentation metrics

563 Figures 5a and 5b show scatter plots for segmentation-precision and segmentation-
564 recall for the $\alpha = 0.5$ *true positive* and *split* components in all 140 masks of
565 the test images, respectively. These correspond to their respective columns of
566 (a non-summarized version of) Table 3 with black and white dots representing
567 the values of FCN-MN and SW detection models, respectively. The position of
568 each dot in the plot corresponds to the mean segmentation-precision and mean
569 segmentation-recall over all the true positive components (splitted components,
570 respectively) of the masks produced by the detection model associated to that
571 dot. The standard deviation of the recall (precision) is shown as a horizontal
572 (vertical) bar.

573 In Figure 5a (true positives), one can observe that all black dots (FCN-MN)
574 are clustered in the upper-right corner of the graph, enclosed by a minimum
575 precision and recall above 70%, while the white dots (SW), also clustered in the
576 upper-right corner, are enclosed in a slightly smaller minimum precision and
577 recall of 50% and 65%, respectively.

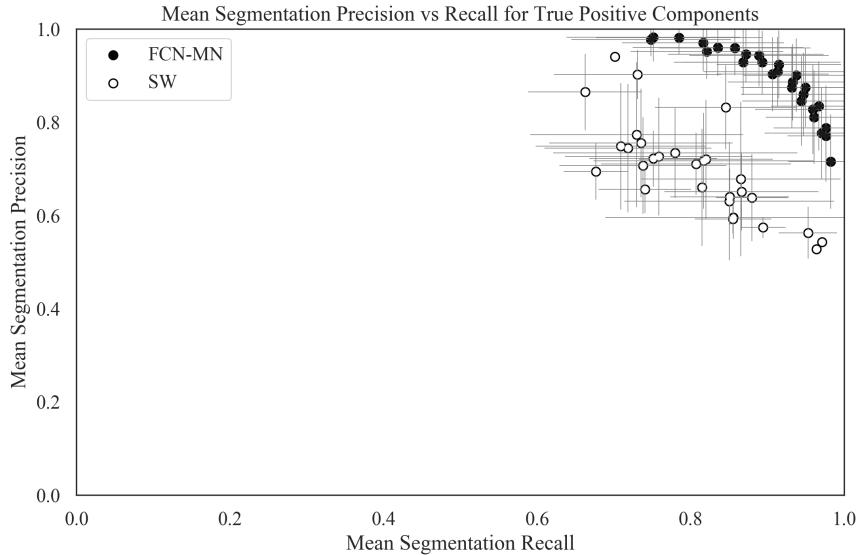
578 In Figure 5b (splits), one can observe a rather different scenario, with FCN-
579 MN showing split components with precisions as large as 100% but small recalls
580 spanning the range of 10% to a maximum 60%, while SW is showing the opposite
581 trend recalls ranging from a low 15% to a maximum of 100% but small preci-

582 sions all below 60%. When read properly, these results show, again, a better
583 performance of FCN-MN against SW, with the former resulting in small splits
584 (low recall) but mostly within the enclosure of the true bud (large precision),
585 while the latter resulting in components reaching beyond the enclosure of the
586 true bud (low precision).

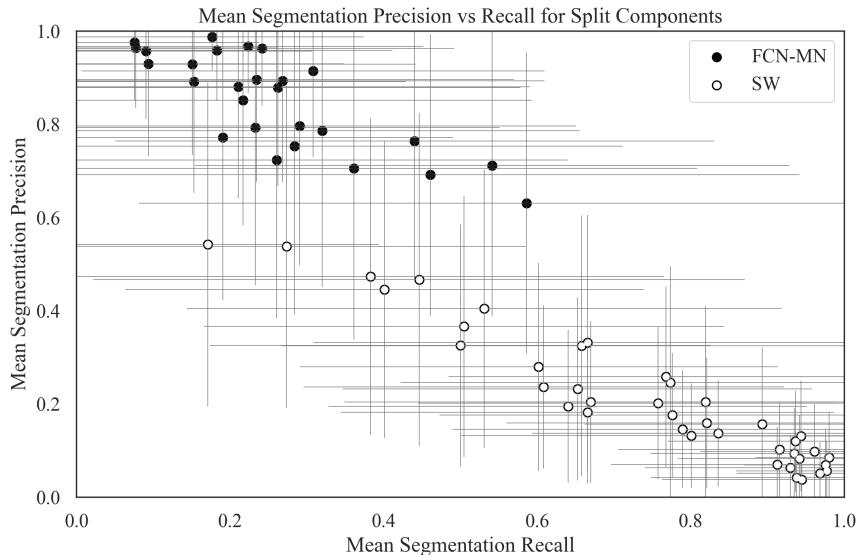
587 Figure 6 show a graphical representation of the segmentation results for the
588 false alarm components, the NA for each of the 27 models of FCN-MN and each
589 of the 40 models of SW, i.e., for each cell in the one-before-last column of (a
590 non-summarized version of) Table 3. results are grouped in two histograms, one
591 for the FCN-MN detection models (black) and one for the SW models (white).
592 Bars in the histogram represent the proportion of detection models whose mean
593 NA (over all false alarm components of all images) falls within the bin interval.
594 The more concentrated to the left the better the algorithm, as this indicates that
595 more detection models for that algorithm resulted in smaller NA (on average).
596 When compared to the histogram of SW, one can observe that the histogram for
597 FCN-MN is considerably more concentrated towards the left, with all FCN-MN
598 models concentrated in a single bar at the left-most interval of [0.0, 1.0]. For
599 SW, the situation is rather different with bars at intervals as far to the right as
600 [57.0, 58.0], that is, detection models with areas as large as 58 times the bud
601 area. These high values correspond to SW models with large window sizes, e.g.,
602 1000px, that for low thresholds are classified as bud patches, rendering all its
603 pixels as bud pixels.

604 3.2.3. Detailed analysis of localization metrics

605 To conclude, this subsection presents a graphical representation of the local-
606 ization results reported in Table 3, that is, the *normalized distance* (ND) only
607 for the $\alpha = 0.5$ false alarms. Figure 7 summarizes the ND values reported in
608 the corresponding column of the (non-summarized version of) Table 3 in the
609 form of two histograms, one for FCN-MN (black) and one for SW (white). Bars
610 in the histogram represent the proportion of detection models (27 for FCN-MN
611 and 40 for SW) whose mean ND falls within the bin interval. The more concen-
612 trated to the left the better the algorithm, as this indicates that more detection
613 models for that algorithm resulted in smaller ND (on average). Here, again,
614 the advantage of FCN-MN over SW is clear, with the histogram for FCN-MN



(a)



(b)

Figure 5: Segmentation Precision-Recall scatterplots reporting the results for FCN-MN and SW in black and white, respectively, with dots representing the segmentation precision and segmentation recall average over all images in the test set (and bars representing standard deviations) with one dot per hyper-parameter configuration (27 for FCN-MN and 40 for SW). In (a) averages were computed over the segmentation precision and recall of all $\alpha = 0.5$ true positive components, while in (b), averages were computed over the segmentation precision and recall of the $\alpha = 0.5$ split components. Recall and precision standard deviations are represented by the horizontal and vertical grey error bars, respectively.

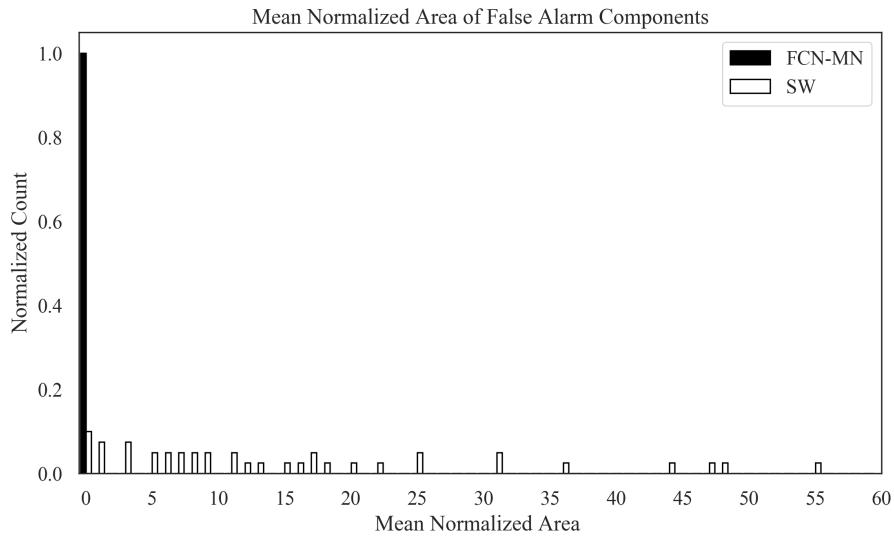


Figure 6: FCN-MN (black bars) and SW (white bars) histograms of the mean normalized area NA of false alarm components with bars representing the proportion of detection models whose mean NA falls within the bin interval.

more concentrated in the left-most part than that of SW, with the FCN-MN histogram running from the $(0, 1]$ to the $(7, 8]$ bin, and the SW histogram running from the $(5, 6]$ towards the $(9, 10]$ bin; and their respective maximums are at $(3, 4]$ and $(7, 8]$, respectively, indicating that most FCN false alarms are at a distance of 3 to 4 bud diameters, while most SW's false alarms are at 7 to 8 bud diameters.

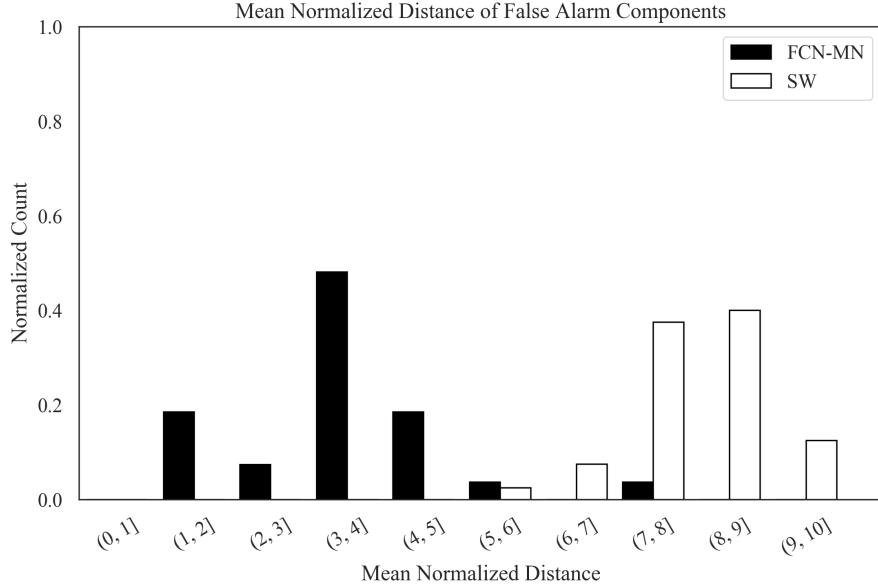


Figure 7: FCN-MN (black bars) and SW (white bars) histograms of mean normalized distance ND over all false alarm components with bars representing the proportion of detection models whose mean ND falls within the bin interval.

621 4. Discussion and Conclusions

622 This section discusses the results obtained by the proposed approach in the
 623 context of the problem of grapevine bud detection and its impact as a tool for
 624 measuring viticultural variables of interest. The discussion is complemented
 625 with some highlights of the most important conclusions together with some
 626 potential lines of future work.

627 This work introduces FCN-MN, a Fully Convolutional Network with Mo-
 628 bileNet architecture for the detection of grapevine buds in 2D images captured
 629 in natural field conditions in winter (i.e., no leaves or bunches) containing a
 630 maximum of one bud.

631 The experimental results confirmed our main hypothesis: that the detection
 632 quality achieved by FCN-MN is improved over the *sliding windows* detector
 633 (SW) in all three detection aspects: segmentation, correspondence identification
 634 and localization. Being SW the best bud detector known to these authors, one
 635 can conclude that FCN-MN is a strong contender in the state-of-the-art for bud
 636 detectors. However, improving over SW is not enough to prove the practical

637 impact of the proposed detection algorithm, as it may still result in a bud
638 detector with limitations for addressing the *quality* requirements of practical
639 measurements of bud-related variables (c.f. in Table 1).

640 Quality performance could be assessed by the metrics reported in Table 3.
641 In the best case, when correct detections (true positives) are considered to
642 be those that overlap true buds with an *IoU* of at least 0.5, FCN-MN shows
643 a detection-precision and detection-recall of 88.6% and 95.0%, respectively, a
644 mean (and standard deviation) segmentation-precision and segmentation-recall
645 for true positives of 98.2%(5.1) and 98.2%(5.5), respectively; and for splits
646 98.8%(7.2) and 58.6%(50.5), respectively. For false alarms, it shows a minimum
647 *NA* of 0.04(0.09) and a minimum *ND* of 1.10(0.65). However, each of these
648 best cases occur for different FCN-MN detectors. A better assessment must
649 be conducted for a single detector. A balanced choice is the detector FCN-
650 MN_{16s}^{0.6}. This detector reaches detection-precision and detection-recall of 88.6%
651 and 88.6%, respectively, meaning than only 11.4% of all the detected connected
652 components over all test images are false positives, and that only 11.4% of all
653 true buds could not be detected (i.e., are false negatives). As expected, for splits
654 it resulted in small mean *IoU* of 18.6%(19.5), corresponding to a mean segmen-
655 tation precision of 89.3%(21.7) and a mean segmentation recall of 26.9%(34.1).
656 A small recall or small precision is expected for a small *IoU*, but a large preci-
657 sion and small recall is preferred as it corresponds to small components mostly
658 within the true bud. True positives resulted in mean *IoU* of 83.1%(9.4), con-
659 siderably larger than the minimum threshold of 0.5, with correspondingly large
660 mean segmentation precision and recall of 92.8%(6.7) and recall of 89.3%(10.2),
661 respectively. The false alarm results for this detector showed an *NA* = 0.08
662 and *ND* = 1.1, showing that these components are rather small covering only
663 an area that is 8% in size of the total bud area (on average) and distant to the
664 true bud by only 1.1(0.65) diameters, on average.

665 With one select best model it is now possible to assess the impact of its good
666 metrics on the quality for the measurement of different bud related variables.
667 For brevity, this point is discussed for three variables selected from Table 1: *bud*
668 *number*, *bud area*, and *internode length*.

669 The case of *bud number*, for example, requires identifying correspondences for

buds in the scene, so its quality will be impacted only by the metrics of detection precision and recall (88.6% and 88.6%, respectively). To evaluate this impact, it is considered that a plant has approximately 240 buds on average. The number of buds per plant depends on many factors, such as training system, grape variety, type of treatment, time of year, among others, so this value is defined as indicative to achieve an approximate analysis. For this particular case, a detection-precision of 88.6% would result in 27 buds counted in excess per plant, while a recall of 88.6% would result in the omission of 27 buds in the count; resulting in a bud count that exactly matches the true count. This suggests that this particular hand-picked model may be the best for the specific task of measuring bud number, as long as these values of precision and recall remain similar. However, even if it is indeed the case that these good results generalize to practical situations, the approach still presents some practical limitations for the measurement of bud number. Namely, it is incapable of associating counts of the same bud appearing in two different images, limiting the scalability to massive measurements of the bud count of a plant or plot.

The second variable of interest considered is *bud area*, where, in addition to identifying correspondences for the buds of a scene, it is necessary to segment it to estimate its area in pixels. Correspondence identification analysis is analogous to bud counting, so now only segmentation metrics are discussed. A thorough error analysis requires an estimation of both *false negative pixels* or fnx corresponding to undetected bud pixels; and *false positive pixels* or fpx corresponding to non-bud detected pixels. These, however, are impossible to compute exactly from existing segmentation metrics. False negative pixels are encoded in the segmentation-recalls of *true positives* and *splits*, corresponding to the sum of their complements. However, we have only their segmentation-recall means, $R_S^{TP} = 89.3\%$ and $R_S^S = 26.9\%$, which adds to more than 100%. As an approximation we assume $fnx = 0$. False positive pixels are encoded in the normalized area of *false alarms* and the segmentation-precisions of *true positives* and *splits*. Precision, however, is normalized by the detected area, not the true bud area. For the lack of a better solution, one can approximate fpx ignoring this distinction, computing fpx as the sum of 8%, the (mean) false alarms NA , with 7.2% and 10.7%, the complements of the mean segmentation-precision of

703 true positives and splits ($P_S^{TP} = 92.8\%$ and $P_S^S = 89.3\%$, respectively). This
704 results in an approximate fpx equal to 25.9%, which given $fnx = 0$, corre-
705 sponds to the total error. For illustrative purposes, we see that this error is
706 equivalent to the precision error resulting from measuring the area of a bud
707 with a caliper. If we assume that the shape of a bud fits a circle, and that the
708 typical diameter of a bud is 5 mm, the resulting area is $19.63mm^2$. Since a
709 caliper has an accuracy of 0.1mm, the area precision error would be $\pm 1.7mm^2$,
710 equivalent to 8.6% of the total area, to which one should add the error of manual
711 measurement resulting from assuming a circular bud shape. From these, one
712 can conclude that, modulo the approximations, both the FCN-MN and caliper
713 errors are equivalent.

714 As in the case of counting, these good results in measurement precision are
715 limited to achieve a practical use of this type of measurement because it is
716 impossible to automatically associate area measurements of the same bud in
717 two different images, making it difficult to systematically measure this variable
718 for the buds of a plant or plot. Furthermore, in this case, the areas obtained
719 are in pixels, which need to be converted into length or area magnitudes.

720 Finally, the case of *internode length* is considered, estimated by the dis-
721 tance between buds of the same branch (by the closeness between buds and
722 nodes), which involves the operations of correspondence identification and lo-
723 calization. Again, correspondence identification analysis is analogous to bud
724 counting, which in this case will result in the reporting of more than one dis-
725 tance due to the detection of more than one component per bud. Among these
726 distances, it is understood that the worst case can occur between two false
727 alarms when they are at the farthest side to the other bud, at a distance ND .
728 On average, ND is 1.1 bud diameters, equivalent to 5.5mm after taking a typ-
729 ical vine bud diameter to be 5mm, resulting in a 7.3% error in estimating the
730 distance between buds/nodes by taking the typical bud distances to be approxi-
731 mately 15cm. An important limitation of our approach for achieving a practical
732 use of this measurement is the possibility of determining when two buds are
733 on the same branch, which requires knowledge of the plant structure. Further-
734 more, with our method, only the distance projected in the image plane could
735 be measured, which can arbitrarily differ from the actual distance in 3D.

736 The greatest impact errors occur because of the excess or omission of con-
737 nected components, with the excess error exacerbated by the fact of associating
738 detected buds with individual connected components. A possible improvement
739 to mitigate these errors would be to apply some post-processing. One such
740 post-processing is *spatial clustering* of connected components grouping them by
741 proximity. One could expect this to improve the results based on the small
742 areas of split and false alarm components. First, due to the closeness of the
743 false alarms to the true bud (small ND) –as well as the splits and true positive
744 components (overlapping with it)–, and the fact that true buds in real plants
745 are typically tens or even hundreds of bud diameters apart, one could expect
746 that a simple spatial clustering of the components would connect all of them
747 together as a single, and correct, bud detection. Second, due to their small area
748 –if clustered together– the false alarm components would only slightly reduce
749 segmentation precision.

750 Another possible post-processing would be to rule out small connected com-
751 ponents, for example, whose area in pixels normalized to the total detected area
752 (sum of the areas of all connected components) is less than a certain threshold.
753 Improvements could be expected with this post-processing, since the results in
754 this work show that false alarms present small areas in relation to the true bud.
755 Lastly, connected component filters could be considered based on plant struc-
756 ture, for example, ruling out connected components that are far away from (or
757 do not overlap with) branches.

758 One could also consider in future works some improvements to overcome the
759 limitations for practical use mentioned above: (i) no associations between plant
760 parts of different images, (ii) distance and area measurements in pixels, (iii)
761 only 2D geometry, (iv) lack of knowledge of underlying plant structure, and (v)
762 need of images with no leaves.

763 One could also extend to buds the work of [Santos et al. \(2020\)](#) that addresses
764 limitation (i) for grape bunches. Limitation (ii) could be easily addressed by
765 adding to the visual scene some marker with known dimensions. This, how-
766 ever, requires such a marker in every image captured, a problem that could be
767 overcome by first producing a calibrated 3D reconstruction of the scene, i.e., a
768 3D reconstruction calibrated with a single marker in one of its frames ([Hartley](#)

769 and Zisserman, 2003; Moons et al., 2009). In this way, every 2D image could
770 be calibrated against the 3D model, omitting the need for a marker. In addition,
771 a 3D reconstruction of the scene could address limitation (iii) by locating
772 the detected buds in 3D space, following, for instance, the approach taken by
773 Díaz et al. (2018). Finally, a solution to limitations (iv) and (v) would require
774 an integrated approach involving the detection in 3D of branches and leaves,
775 respectively.

776 To end, future research could examine a comprehensive evaluation of the
777 FCN-MN detector over images with multiple bud cases. This challenging task
778 involves the creation of a new corpus, the training of new FCN-MN models, and
779 the systematic evaluation of experiments. Such work may bring about a greater
780 impact to FCN-MN, by being able to validate its performance over a broader
781 range of practical cases that may take place in real vineyards.

782 Acknowledgments

783 This work was funded by the Argentinean *Universidad Tecnológica Nacional*
784 (UTN), the National Council of Scientific and Technical Research (CONICET),
785 and the National Fund for Scientific and Technological Promotion (FONCyT).

786 References

- 787 Berenstein, R., Shahar, O.B., Shapiro, A., Edan, Y., 2010. Grape clusters
788 and foliage detection algorithms for autonomous selective vineyard sprayer.
789 Intelligent Service Robotics 3, 233–243.
- 790 Bramley, R.G., 2009. Lessons from nearly 20 years of precision agriculture
791 research, development, and adoption as a guide to its appropriate application.
792 Crop and Pasture Science 60, 197–217.
- 793 Collins, C., Wang, X., Lesefko, S., De Bei, R., Fuentes, S., 2020. Effects of
794 canopy management practices on grapevine bud fruitfulness. OENO One 54,
795 313–325.
- 796 Diago, M.P., Correa, C., Millán, B., Barreiro, P., Valero, C., Tardaguila, J.,
797 2012. Grapevine yield and leaf area estimation using supervised classification

- 798 methodology on rgb images taken under field conditions. Sensors 12, 16988–
799 17006.
- 800 Díaz, C.A., Pérez, D.S., Miatello, H., Bromberg, F., 2018. Grapevine buds
801 detection and localization in 3d space based on structure from motion and 2d
802 image classification. Computers in Industry 99, 303–312.
- 803 Garcia-Garcia, A., Orts-Escalano, S., Oprea, S., Villena-Martinez, V., Martinez-
804 Gonzalez, P., Garcia-Rodriguez, J., 2018. A survey on deep learning tech-
805 niques for image and video semantic segmentation. Applied Soft Computing
806 70, 41–65.
- 807 Grimm, J., Herzog, K., Rist, F., Kicherer, A., Töpfer, R., Steinhage, V., 2019.
808 An adaptable approach to automated visual detection of plant organs with
809 applications in grapevine breeding. Biosystems Engineering 183, 170–183.
- 810 Han, D., 2013. Comparison of commonly used image interpolation methods,
811 in: Proceedings of the 2nd international conference on computer science and
812 electronics engineering, Atlantis Press.
- 813 Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision.
814 Cambridge university press.
- 815 Herzog, K., Kicherer, A., Töpfer, R., 2014a. Objective phenotyping the time of
816 bud burst by analyzing grapevine field images, in: XI International Confer-
817 ence on Grapevine Breeding and Genetics 1082, pp. 379–385.
- 818 Herzog, K., et al., 2014b. Initial steps for high-throughput phenotyping in
819 vineyards. Australian and New Zealand Grapegrower and Winemaker , 54.
- 820 Hirano, Y., Garcia, C., Sukthankar, R., Hoogs, A., 2006. Industry and ob-
821 ject recognition: Applications, applied research and challenges, in: Toward
822 category-level object recognition. Springer, pp. 49–64.
- 823 Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T.,
824 Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural
825 networks for mobile vision applications. arXiv preprint arXiv:1704.04861 .
- 826 Jaccard, P., 1912. The distribution of the flora in the alpine zone. 1. New
827 phytologist 11, 37–50.

- 828 Kahng, M., Andrews, P.Y., Kalro, A., Chau, D.H.P., 2017. A ctiv is: Visual
829 exploration of industry-scale deep neural network models. *IEEE transactions*
830 on visualization and computer graphics
- 24, 88–97.
- 831 Kaymak, Ç., Uçar, A., 2019. A brief survey and an application of semantic
832 image segmentation for autonomous driving, in: *Handbook of Deep Learning*
833 Applications. Springer, pp. 161–200.
- 834 Kornblith, S., Shlens, J., Le, Q.V., 2019. Do better imagenet models trans-
835 fer better?, in: *Proceedings of the IEEE conference on computer vision and*
836 *pattern recognition*, pp. 2661–2671.
- 837 Lampert, C.H., Blaschko, M.B., Hofmann, T., 2008. Beyond sliding windows:
838 Object localization by efficient subwindow search, in: *2008 IEEE conference*
839 *on computer vision and pattern recognition*, IEEE. pp. 1–8.
- 840 Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian,
841 M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I., 2017. A survey on
842 deep learning in medical image analysis. *Medical image analysis* 42, 60–88.
- 843 Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for
844 semantic segmentation, in: *Proceedings of the IEEE conference on computer*
845 *vision and pattern recognition*, pp. 3431–3440.
- 846 Matese, A., Di Gennaro, S.F., 2015. Technology in precision viticulture: A state
847 of the art review. *International journal of wine research* 7, 69–81.
- 848 May, P., 2000. From bud to berry, with special reference to inflorescence and
849 bunch morphology in *vitis vinifera* l. *Australian Journal of Grape and Wine*
850 *Research* 6, 82–98.
- 851 Moons, T., Van Gool, L., Vergauwen, M., 2009. *3D Reconstruction from Mul-*
852 *tiple Images: Principles*. Now Publishers Inc.
- 853 Ning, C., Zhou, H., Song, Y., Tang, J., 2017. Inception single shot multibox
854 detector for object detection, in: *2017 IEEE International Conference on*
855 *Multimedia & Expo Workshops (ICMEW)*, IEEE. pp. 549–554.

- 856 Noyce, P.W., Steel, C.C., Harper, J.D., Wood, R.M., 2016. The basis of defolia-
857 tion effects on reproductive parameters in *vitis vinifera* l. cv. chardonnay lies
858 in the latent bud. *American Journal of Enology and Viticulture* 67, 199–205.
- 859 Nuske, S., Achar, S., Bates, T., Narasimhan, S., Singh, S., 2011. Yield estima-
860 tion in vineyards by visual grape detection, in: 2011 IEEE/RSJ International
861 Conference on Intelligent Robots and Systems, IEEE. pp. 2352–2358.
- 862 Pan, S.J., Yang, Q., 2009. A survey on transfer learning. *IEEE Transactions on*
863 *knowledge and data engineering* 22, 1345–1359.
- 864 Pérez, D.S., Bromberg, F., Diaz, C.A., 2017. Image classification for detection
865 of winter grapevine buds in natural conditions using scale-invariant features
866 transform, bag of features and support vector machines. *Computers and*
867 *electronics in agriculture* 135, 81–95.
- 868 Rudolph, R., Herzog, K., Töpfer, R., Steinhage, V., 2018. Efficient identi-
869 fication, localization and quantification of grapevine inflorescences in un-
870 prepared field images using fully convolutional networks. *arXiv preprint*
871 *arXiv:1807.03770* .
- 872 Sánchez, L.A., Dokoozlian, N.K., 2005. Bud microclimate and fruitfulness in
873 *vitis vinifera* l. *American Journal of Enology and Viticulture* 56, 319–329.
- 874 Santos, T.T., de Souza, L.L., dos Santos, A.A., Avila, S., 2020. Grape detection,
875 segmentation, and tracking using deep neural networks and three-dimensional
876 association. *Computers and Electronics in Agriculture* 170, 105247.
- 877 Seng, K.P., Ang, L.M., Schmidtko, L.M., Rogiers, S.Y., 2018. Computer vision
878 and machine learning for viticulture technology. *IEEE Access* 6, 67494–67510.
- 879 Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for
880 semantic segmentation. *IEEE transactions on pattern analysis and machine*
881 *intelligence* 39, 640–651.
- 882 Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation
883 for deep learning. *Journal of Big Data* 6, 60.

- 884 Siam, M., Gamal, M., Abdel-Razek, M., Yogamani, S., Jagersand, M., 2018.
885 Rtseg: Real-time semantic segmentation comparative study, in: 2018 25th
886 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 1603–
887 1607.
- 888 Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-
889 scale image recognition. CoRR abs/1409.1556.
- 890 Tardaguila, J., Diago, M., Blasco, J., Millán, B., Cubero, S., García-Navarrete,
891 O., Aleixos, N., 2012. Automatic estimation of the size and weight of
892 grapevine berries by image analysis, in: Proc. CIGR AgEng.
- 893 Tardáguila, J., Diago, M.P., Millan, B., Blasco, J., Cubero, S., Aleixos, N., 2012.
894 Applications of computer vision techniques in viticulture to assess canopy
895 features, cluster morphology and berry size, in: I International Workshop on
896 Vineyard Mechanization and Grape and Wine Quality 978, pp. 77–84.
- 897 Tarry, C., Wspanialy, P., Veres, M., Moussa, M., 2014. An integrated bud
898 detection and localization system for application in greenhouse automation,
899 in: 2014 Canadian Conference on Computer and Robot Vision, IEEE. pp.
900 344–348.
- 901 The Australian Wine Research Institute, a. Viticare on Farm Trials - Manual
902 3.1: Measuring Fruit Quality. 1 ed. The Australian Wine Research Institute.
903 Accessed August 2020.
- 904 The Australian Wine Research Institute, b. Viticare on Farm Trials - Manual
905 3.3: Vine Health. 1 ed. The Australian Wine Research Institute. Accessed
906 August 2020.
- 907 Tilgner, S., Wagner, D., Kalischewski, K., Velten, J., Kummert, A., 2019. Multi-
908 view fusion neural network with application in the manufacturing industry,
909 in: 2019 IEEE International Symposium on Circuits and Systems (ISCAS),
910 IEEE. pp. 1–5.
- 911 Whalley, J., Shanmuganathan, S., 2013. Applications of image processing in
912 viticulture: A review .

- 913 Whelan, B., McBratney, A., Viscarra Rossel, R., 1996. Spatial prediction for
914 precision agriculture, in: Proceedings of the Third International Conference
915 on Precision Agriculture, Wiley Online Library. pp. 331–342.
- 916 Xu, S., Xun, Y., Jia, T., Yang, Q., 2014. Detection method for the buds
917 on winter vines based on computer vision, in: 2014 Seventh International
918 Symposium on Computational Intelligence and Design, IEEE. pp. 44–48.
- 919 Zhao, F., Rong, D., Liping, L., Chenlong, L., 2018. Research on stalk crops
920 internodes and buds identification based on computer vision. MS&E 439,
921 032080.