Published in the ICCV 8th workshop on Computer Vision in Plant Phenotyping and Agriculture, Paris, October 2023

Efficient Grapevine Structure Estimation in Vineyards Conditions

Theophile Gentilhomme[1] Michael Villamizar[1] Jerome Corre[2] Jean-Marc Odobez[1]

[1]Idiap Research Institute, [2]3D2cut SA

{theophile.gentilhomme, michael.villamizar, odobez}@idiap.ch, jerome@3d2cut.com

Abstract

Developing computer vision systems for agricultural tasks that work in real-world conditions and in real time is challenging, especially if they need to be deployed on embedded devices, such as tablets or augmented reality glasses. In this paper, we present an efficient deep-learning approach for the estimation of grapevine structure in natural conditions with the aim of assisting vinemakers in some decision-making activities like grapevine pruning. Specifically, we propose a lightweight network for detecting nodes and branches in images which are then used to recover the tree structure.

Our approach is validated on the publicly available 3D2Cut dataset. Compared to the ViNet method [11], we demonstrate computational performance while preserving the high accuracy of its predictions. Furthermore, we created a new dataset to train our workflow in real vineyard conditions without an artificial background. We demonstrate that we can obtain remarkable results in real and challenging conditions while being efficient.

1. Introduction

Motivation. Some agricultural tasks require a high precision and expert knowledge to be performed and are often labor-intensive and time-consuming. One such case is grapevine pruning, where making precise cuts in the right places is critical. Badly placed cuts can damage the vine's vascular system, disrupting the flow of sap and nutrients, which can lead to diseases [3, 7].

Several attempts have been made to automate this challenging task [2, 6, 10, 16], including the use of complex robotic approaches, but they do not meet the level of precision required to maintain a vineyard at its top productivity and quality. In particular, deciding where to cut requires a precise understanding of the tree structure, of the location and orientation of buds, and so on.

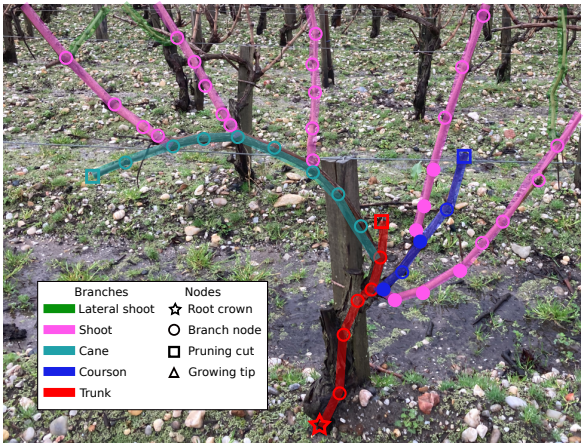

Figure 1: Our approach detects nodes and branches and reconstructs the plant structure from images without the need for artificial backgrounds. Relevant nodes used in the Courson metric are represented by filled dots, as described in Section 5.

Focusing on an augmented reality solution, authors in [11] proposed a deep-learning vision system, called ViNet, that precisely detects the different parts of a Guyot vine and then reconstructs the plant structure using a resistivity graph. Despite the good results obtained (a recall and precision of 90% and 95%, respectively), this system was only applied to images with artificial blue or white backgrounds from a new and publicly available dataset comprising thousands of grapevine images (3D2cut dataset). Furthermore, processing one image on the CPU takes 8 seconds, which does not meet the close to real-time execution requirements for its use in embedded systems.

In this work, we take a step towards real-world applications and propose an efficient deep-learning approach

to estimate the grapevine structure in vineyard conditions without artificial backgrounds (see Figure 1). This approach comprises a lightweight network architecture that was conceived to significantly reduce computational time while yielding precise predictions (nodes and branches).

As a result, the proposed approach brings greater productivity and opportunities to agricultural tasks such as pruning vineyards since non-expert workers can perform the task assisted by the deep-learning vision system, which can be embedded into a hand-held device or glasses. This approach can also be exploited for other tasks, such as grapevine phenotyping [21], since it can extract phenotypic data like the size and morphology of the plant.

Approach and contributions. Our approach takes inspiration from the ViNet network. Different architectural designs were investigated to be computationally more efficient while preserving the accuracy of predictions. In particular, our network, named EViNet, uses inverted residual and linear bottleneck blocks to reduce the number of parameters and have a lightweight backbone for feature extraction [27].

The extraction of the tree structure is done in two main steps: one for the detection and recognition of nodes and segments of the plant using the proposed network, and the second one for the association of the nodes and estimation of the structure of the vine.

In this work, we focus on improving the first step, which is decisive for good performance and dominates the runtime of our vision system. In addition to efficiency, and in contrast to [11], we also investigate the performance of our method in challenging vineyard conditions in which the background includes parts of other vines, and lighting and viewpoint variations are present. In summary, our contributions are as follows:

- we propose an efficient network (EViNet) based on lightweight convolutional blocks and relevant architectural designs, which is significantly more efficient than ViNet;
- we train and evaluate our approach on a newly created dataset called RealGuyot, which contains a set of carefully annotated vineyard images with natural backgrounds;
- we study and conduct multiple experiments to assess different configurations of EViNet for grapevine structure estimation.

Overview. The paper is organized as follows. In the next section, we present some related works in the state of the art. Section 3 describes the 3D2cut dataset [11] used for training and testing our approach, as well as, our new set of vineyard images (free of artificial backgrounds) with their corresponding annotations. In Section 4, we introduce the proposed network and describe its main components in detail. In Section 5, EViNet is evaluated and contrasted in terms of efficiency and accuracy against ViNet. Future work and conclusions are commented on Section 6.

2. Related work

Vine structure estimation. In recent years, we have seen an increase of deep-learning methods applied to agricultural tasks. Deep networks have been used for tasks such as fruit detection [1, 25, 20], crop forecasting [23, 8], or plant disease detection [9, 26]. However, only a few works have been proposed for grapevine tasks, such as [5], which performs image segmentation of some parts of the plant, although the plant structure was not recovered. In [15], deep networks for object detection were used to detect visible segments of full foliage grapevine canopies via the prediction of multiple and overlapping bounding boxes.

Authors in [11] were the first to address the vine tree structure estimation problem. Their system, ViNet, first performs a detection step to identify the nodes and determine their spatial relationships (branches), while in a second step, a node association step is done to reconstruct the entire plant structure from the detected nodes and branches using a resistivity graph and a shortest-path algorithm. The detection is based on a stacked-hourglass network [22] that predicts feature maps, representing the location of the different nodes, and vector fields that model branches and provide information about the connection and direction between nodes. This technique is commonly employed in the field of human pose estimation to detect body keypoints and limbs [4, 19, 17, 18]. A great disadvantage of this work is its high computational cost, which makes it unfeasible to deploy on embedded devices for real-world applicability since, although the second step is efficient, the detection step takes much longer (over 7 seconds per image).

Model optimization. There are several techniques that can be used for reducing the computational complexity of a network and its inference time. One of them is *quantization* [29] which can be applied to a neural network without modifying its architecture. This technique consists of lowering the precision of the network's weights, for example, by replacing 32-bit floating-point numbers with 8-bit integers. This results in significant reductions in memory usage and improved computational efficiency. However, in early experiments, we found it is challenging to apply this technique to our model in practice because of some technical difficulties.

Knowledge distillation [12] is another technique that involves training a smaller and more computationally efficient network to emulate the behavior of a larger and more complex network, leveraging the knowledge captured by the larger model to improve the efficiency of the smaller one. While this approach can be effective, we decided to follow a network architecture optimization approach in this work. Hence, to improve the efficiency of our network, we

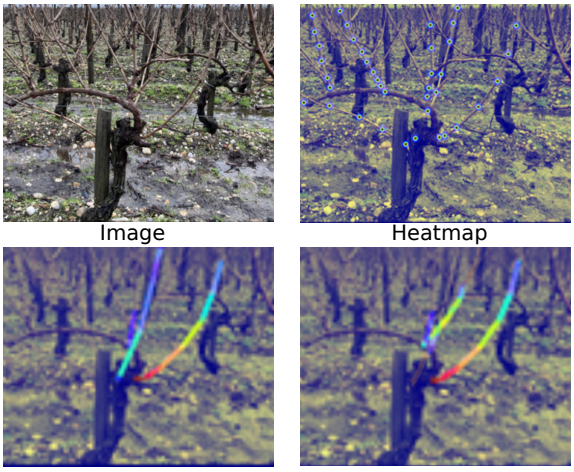

Figure 2: Example of ground-truth heatmap and vector fields characterizing the Shoots nodes and connections. Top-left: input image. Top-right: a heatmap representing the Shoots nodes. Bottom: X-Y vector field components. It is worth noting that these maps were superimposed onto the original image purely for illustrative purposes. That is, the network does not reconstruct the background image.

decided to redesign the hourglass architecture used in [11] by replacing the encoder part with inverted residual blocks from a pretrained MobilenetV2 network [27]. This brings a significant reduction in the computational cost and memory usage of network models [27, 28, 14, 13]. In addition, we studied different processing and network configurations (input/output resolutions and number of feature channels) to find a trade-off between processing speed and accuracy.

3. Datasets and annotations

In this section, we describe the 3D2cut dataset and its annotation procedure [11], as well as the newly created dataset (RealGuyot) for grapevine images in real-world conditions.

Annotations. Nodes and branches are annotated for single Guyot vine as shown in Figure 1. The vine's oldest part is called the Trunk (red), and it branches into the Courson (blue) and Cane (turquoise). The Cane is a two-year-old branch that typically bends and attaches to supporting wires, carrying branches like Shoots from the current year.

For grapevine pruning, the Cane is removed and replaced with the second Shoot growing from the Courson. The first Shoot then becomes the new Courson for the next year, and is pruned to have at least two buds. Thus, an accurate prediction of nodes and branches in the Courson area is crucial.

Ground-truth encoding. From the positions and connections given by the annotations, heatmaps and vector fields are generated to supervise the training of the network, as illustrated in Figure 2. There are a ground-truth heatmap for each node type, and a vector field for each branch type. For each node type, gaussian blobs are used to encode the positions of all node for this type, with their spread defined by the standard deviation σ_{gt}. Vector fields are encoded as two-channel maps, with each pixel encoding the presence (norms) or not of a branch of that type, and the direction encodes the direction of the branch at this point.

Datasets. To evaluate the performance of EViNet, we use two datasets of grapevine images. The first one is the 3D2cut dataset, which allows us to make direct comparisons between our proposed network and the ViNet network. Specifically, this dataset has 1255 images sampled across different vineyards. To ensure that the plant being processed is the main focus and to avoid the presence of other plants in the background, artificial backgrounds were set during data collection.

The second dataset is a small set of grapevine images with natural backgrounds that we call the RealGuyot dataset. It consists of 274 images that we annotated in a manner similar to the 3D2cut dataset. However, unlike this latter one, occluded nodes were systematically annotated in the RealGuyot dataset. Furthermore, the testing images in RealGuyot contain more complex vine structures, such as Courson without shoots or no Courson at all. This results in a more challenging and realistic dataset for benchmarking different models. Figure 5 shows some example images that evidence the difficulty of extracting the grapevine structure in real conditions without artificial backgrounds.

4. Methods

In this section, we present the different networks that have been assessed and review the association method used to reconstruct the structure of the grapevine. Regarding networks, firstly, we briefly describe ViNet (Sec. 4.1) and then we present some configuration changes done in this work to make it more efficient (Sec. 4.2). Ultimately, in Sec. 4.3, our EViNet network is introduced in more detail.

4.1. Baseline: ViNet

Our reference network is ViNet [11], which relies on a stacked hourglass network [22]. It takes as input the image and predicts a stack of node and branch heatmaps at one fourth of the input resolution, see Figure 3. Specifically, the network comprises a feature extractor, two stacked hourglass subnetworks, and a final convolutional block for prediction refinement. The feature extractor extracts deep features which are then used as input in the hourglass subnetworks for predicting the nodes and branches. Each hourglass is a U-shape network architecture [24], consisting of convolutional blocks, maxpooling and upsampling layers, and lateral skip connections to process features at multiple resolutions. Refer to table 1 for the parameters of this model.

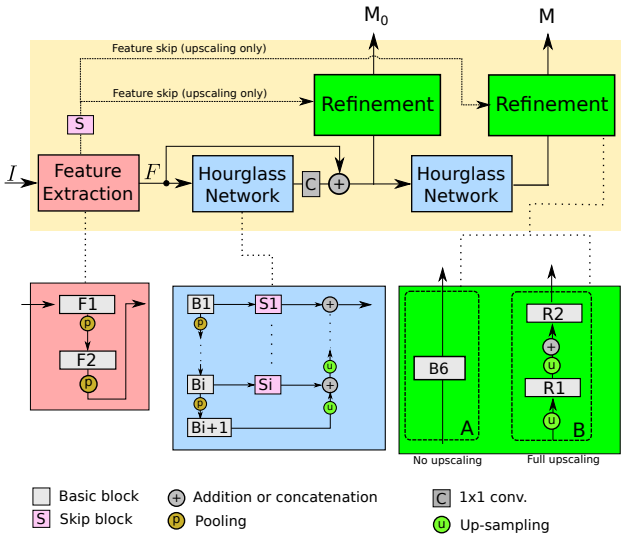

Figure 3: Generic representation of the different architectures evaluated in this work. All includes (1) a feature extractor which downscales and extracts deep features from the input image, (2) stacked hourglasses which process the features at different resolutions, and (3) a refinement step, which produces the final outputs. The refinement step can have different configurations, such as upscaling the features back to the input resolution or producing the output at the hourglass resolution.

4.2. Adapting ViNet for faster inference

In this work, we investigate some setup changes and network modifications to make ViNet faster. Particularly, we tested three configurations that are described below. Details of them are provided in Table 1.

ViNet/F128. The goal of this configuration is to make the network smaller (*i.e.*, with much fewer parameters) to be more efficient. Specifically, we use 128 feature channels across the hourglass subnetworks as opposed to the 256 channels used in ViNet. This yields a network model with 3.3 million parameters which represents a reduction of 75% with respect to the original model (13 million parameters).

ViNet/R512. This configuration simply downscales the resolution of the input image to reduce the computational cost since the size of the input data is smaller. It is a common strategy used in computer vision algorithms to go faster. Particularly, we downscale the input size from 1024×1024 pixels to 512×512 pixels. As a consequence, the output prediction resolution is 128×128 pixels due to the network architecture, which shrinks the resolution by a quarter.

ViNet/R512U. As reducing the output resolution to 128×128 pixels (ViNet/R512) can lead to suboptimal performance since prediction heatmaps are less precise (coarse predictions), we opted to upscale the output of the second hourglass subnetwork to match the input image resolution (512×512 pixels) to obtain more precise predictions. By doing so, we negligibly increase the number of parameters. The upsampling module comprises two upsampling layers along with two convolutional layers with residual connections, see Figure 3.

4.3. Our network: EViNet

In our network, we propose to redesign the feature extractor and the hourglass subnetworks, which are the main components in charge of predicting the nodes and branches and are computationally demanding. Specifically, we replace the residual convolutional blocks with inverted residual blocks as used in MobilenetV2 [27], which have been shown to significantly decrease the number of convolutional layer parameters and speed up feature processing.

For the feature extractor, we use the first 2D convolution layer and the first two bottleneck blocks of MobilenetV2 (see Figure 3), which leads to a downsampling of one-fourth in resolution, similar to the ViNet feature extractor. The remaining 12 bottleneck blocks of MobilenetV2 are used in the encoder part of the hourglass subnetworks. This further downscales the feature maps to 1/32 of the input resolution. Also, three lateral skip connections process the output of the bottleneck blocks to extract features that are combined with features from the decoder. They are composed of one residual block that preserves the same number of channels as in the inverted block for computational efficiency reasons.

In the decoder part of the hourglass subnetworks, we use a succession of upsampling bilinear interpolations, similar to ViNet. Yet, as the number of features varies from one level to another, concatenation of the interpolated and skip connection features is applied, followed by a 1x1 convolution layer to rescale the number of feature channels to 128.

As in Section 4.2, we investigate three different configurations for our proposed network, see Table 1. They are described below.

EViNet. It is our default network which takes 1024×1024 input images and produces 256×256 feature maps. As it uses more efficient feature processing blocks, the number of parameters is relatively small (2.6 millions) and consists of an 80% reduction compared to ViNet.

EViNet/R512U. In this configuration, the input image to EViNet is downscaled to a resolution of 512×512 pixels to process the image faster. In addition, the upsampling module, explained in Section 4.2 (ViNet/R512U), is added to the network to obtain more accurate node and branch heatmaps. The number of parameters has slightly increased, from 2.6 to 2.7 millions (see Table 1).

EViNet/C320U. To further speed up the tree structure estimation, we also propose to crop the 512×512 image using a bounding box of size 320×320 pixels centered on the grapevine. This again decreases the size of the input data, focusing on the central part of the tree, which is the most

Network	Input resolution	Output resolution	# Parameters	Upsampling	Runtime (sec.)
ViNet [11]	1024	256	13M		7.3s
ViNet/F128	1024	256	3.3M		2.8
ViNet/R512	512	128	13M		1.6
ViNet/R512U	512	512	13M	✓	3.6
EViNet	1024	256	2.6M		1.2
EViNet/R512U	512	512	2.7M	✓	1.2
EViNet/C320U	512 (320)	512 (320)	2.7M	✓	0.45

Table 1: Network model settings and runtimes.

relevant for grapevine pruning. This configuration also includes the upsampling module which returns heatmaps with a size of 320×320 pixels.

4.4. Association

We follow the method proposed in [11] for node association and tree structure reconstruction. From the generated heatmaps, the image location of all nodes is extracted using a local maximum detection method. The association method first builds a resistivity graph using nearest neighbors to connect nodes with their potential parents. Resistivity weights are assigned to the edges of the graph, where the resistivity value is calculated based on the local alignment between the vector defined by the two points of the edge and the corresponding vector field associated with a potential node connection. In addition, the resistivity value takes into account the distance between the two nodes. It is low when the edge vector is locally aligned with the vector field and the distance between the two nodes is short.

Subsequently, using the resistivity graph, the Dijkstra algorithm is applied to compute the shortest paths (i.e., paths with the least resistivity) from each node to the Root Crown. The connections determined by the shortest paths define the structure of the tree.

It is important to note that the quality of the resulting tree is mainly determined by the accuracy of the predicted nodes and vector fields provided by the network.

5. Experimental results and discussions

Implementation details. For comparison purposes, we follow the training configuration outlined in [11], while adjusting the spread of Gaussian blobs and vector fields to ensure consistency in the conditioning ground truths and being independent of the differences in their resolutions.

Network training. All networks are trained using supervised learning to ensure that the predicted heatmaps and vector fields are highly similar to the corresponding ground-truth annotations. To achieve this, supervision is applied not only at the output of the last hourglass subnetwork but also after the prediction made by the first one. As illustrated in Figure 3-top, this strategy copes with vanishing gradients by providing error gradient feedback at an earlier stage in the network.

Fine-tuning with real-world vineyard conditions. All networks are first trained on the 3D2cut dataset from images with artificial backgrounds. Those networks are then fine-tuned (all weights) with our RealGuyot dataset. The same training configurations are used in both cases.

Inference. During inference, the networks were run on an AMD 2.25 GHz CPU. As runtime, we report the average runtime when doing inference on 100 images.

Evaluation metrics. For evaluation, we use the same metrics as those used in [11]: *AllNodeMetric* and *CoursonMetric*.

AllNodeMetric. This metric focuses only on the identification and recognition of individual nodes and does not take into account their order within a branch. In other words, the sequence of nodes within a branch is not considered. For instance, it evaluates the accuracy of detecting the Shoot nodes in the entire tree, regardless of their order within each branch.

This metric calculates the recall, precision, and F-score of recognized nodes. Precision is the fraction of relevant nodes among all predicted nodes, while recall is the fraction of relevant nodes that were retrieved. F-score combines precision and recall values. Note that this metric requires to associate ground truth nodes with detected ones. To account for the different output resolutions, when calculating true positives, false positives, and false negatives, nodes are paired with the ground truth using a search radius τ_d of 3, 5, and 10 pixels for an output resolution of 128×128, 256×256 and 512×512, respectively. Thus, the same coverage is considered in all cases.

CoursonMetric. In Section 3, we discussed the importance of nodes in the Courson area for decision-making during pruning. Specifically, the branching node of the Courson and the first three nodes of each Shoot growing out of it are crucial (Figure 1). To capture the significance of these nodes and the order of the Shoot nodes in the branch sequence, we use the CoursonMetric in our analysis. It allows us to report specific metrics for the branching node of the Courson, as well as the first, second, and third nodes of the

Shoots. As a result, we not only account for the detection of the nodes themselves, but also, indirectly, for the association process, both steps being essential for obtaining good pruning decisions.

5.1. Results

The networks considered are evaluated on grapevine images with and without artificial backgrounds.

With artificial backgrounds. The performance rates of all networks on the test set of the 3D2cut dataset are given in Table 2. We see that ViNet obtains the highest scores in both metrics, but this is at the expense of using a larger network model (13 million parameters) and working at full image resolution (1024×1024 pixels), which leads to processing an image in 7.3 seconds (see Table 1 and Figure 4). Hence, it is not feasible to deploy it on small devices with processing and memory hardware limitations.

For all the proposed configurations of ViNet, the performance rates for AllNodeMetric are similar or decrease only slightly. However, we observe a substantial decrease for the CoursonMetric, specially for the recall values. For the case of ViNet/F128, the recall drops from 0.74 to 0.67, indicating that the network has difficulties detecting nodes and branches. It is probably because the network was simply compressed (in terms of the number of feature channels) to get a smaller model, which might be an inappropriate procedure without reformulating the full network structure. Nevertheless, this network architecture is more efficient, running at 2.8 seconds per image.

If the input image is downscaled, the speed is increased. This is the case with ViNet/R512 which processes the image at half-image resolution. It runs at 1.8 seconds per image, which is four times faster than working at full image resolution. However, its performance rates go down on the CoursonMetric for both recall and precision values (0.64 and 0.66, respectively). The cause is that the output resolution of heatmaps is small (128 x 128 pixels), which makes it more difficult to detect and distinguish nodes which are close in the image, a situation which is rather common for the Courson nodes. Moreover, this configuration uses the large network model.

Using the upsampling module (ViNet/R512U), the performance scores get higher, both in recall and precision, obtaining an F-score of 0.73. This proves that higher output resolution is necessary to obtain more accurate predictions. The downside is that the computational cost increases as well, processing an image in 3.6 seconds.

Our proposed network presents different performances for its different configurations. The default network (EViNet) attains one of the lowest performance scores in the CoursonMetric (an F-score of 0.67), probably because it does not use the upsampling module. However, it is remarkable that this network processes an image in 1.2 seconds at full image resolution. As well, it is the smallest network model, with 2.6 million parameters. This is thanks to the proposed lightweight network architecture.

Network	AllNodeMetric Rec./Pre./F-score	CoursonMetric Rec./Pre./F-score
ViNet [11]	**0.90 / 0.95 / 0.93**	**0.74 / 0.76 / 0 .75**
ViNet/F128	0.90 / 0.94 / 0.92	0.67 / 0.74 / 0.70
ViNet/R512	0.88 / 0.94 / 0.91	0.64 / 0.66 / 0.65
ViNet/R512U	0.90 / 0.94 / 0.92	0.72 / 0.74 / 0.73
EViNet	0.87 / 0.95 / 0.91	0.65 / 0.68 / 0.67
EViNet/R512U	0.88 / 0.94 / 0.91	**0.73 / 0.78 / 0.75**
EViNet/C320U	0.81 / 0.94 / 0.87	0.67 / 0.74 / 0.70

Table 2: Evaluation of the performances on the 3D2cut dataset by computing both metrics across all node types.

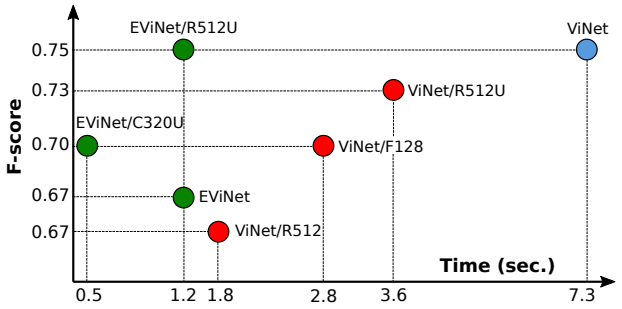

Figure 4: CoursonMetric performance versus efficiency of all networks.

In the case of EViNet/R512U, we obtain competitive performance rates against ViNet. It attains the same F-score (0.75) in the CoursonMetric while being six times faster (running at 1.2 seconds per image). This is achieved by working at half-image resolution and using the upsampling module.

Ultimately, EViNet/C320U is the fastest network configuration (see Figure 4), although its performance in both metrics is low. Particularly, we see a performance drop in the AllNodeMetric when it is compared with other networks. The reason is that this network configuration only processes a part of the image (320×320 crop), which makes it more difficult to capture the full tree structure and recognize the nodes and branches. Nevertheless, it is a good trade-off between efficiency and performance.

Figure 4 plots the efficiency and performance of all networks using the CoursonMetric. We see that the proposed networks are efficient while obtaining good performance.

In vineyard conditions. The performances of the networks on the RealGuyot test set are given in Table 3. Particularly, we are only considering the networks ViNet, ViNet/R512U and EViNet/R512U as they demonstrated higher performance in previous experiments.

Network	AllNodeMetric Rec./Pre./F-score	CoursonMetric Rec./Pre./F-score
ViNet [11]	**0.82 / 0.92 / 0.87**	**0.53 / 0.67 / 0.59**
ViNet/R512U	0.79 / 0.89 / 0.84	0.46 / 0.63 / 0.53
EViNet/R512U	0.75 / 0.90 / 0.82	0.49 / 0.62 / 0.54

Table 3: Evaluation of the performances on the RealGuyot for some networks.

The overall performances of all network models are strongly affected in this testing set. This may be partially explained by slight differences in the annotation process and the condition of the plants contained in the set. Note that EViNet/R512U delivers competitive performance despite having significantly fewer parameters than ViNet/R512U. This suggests that our network optimization strategy is also valid with more complex data. Nevertheless, the ViNet network still gives the best results, especially for the CoursonMetric. This may suggest that high resolution input images have an important effect on the nodes and vector field estimation in vineyard conditions. This should be considered for future work on our optimized networks.

5.2. Qualitative results

Some example images with the output of EViNet/R512U and ViNet are shown in Figure 5.

Although the metrics suggest slightly better performance with ViNet, the qualitative comparison with the EViNet/R512U results reveals a high level of competitiveness between the two networks. Both effectively reconstruct the structure of the plants, with only minor differences observed. This raises the possibility that the CoursonMetric, which was used to evaluate the models, may be too restrictive for a comprehensive assessment of their efficiency. Despite the challenging conditions of real vineyards, both networks yield satisfactory results.

Given that EViNet/R512U is six times faster and five times lighter than ViNet, while still achieving satisfactory results, it makes it a compelling candidate for real-life applications on embedded devices.

6. Conclusions

In this work, we introduced a new deep network with different setups for efficient and accurate estimation of vine plant structures in real vineyard conditions.

Our deep-learning approach leverages the efficient inverted residual blocks used in MobilenetV2 to improve computational efficiency while obtaining accuracy comparable to ViNet. We fine-tuned both EViNet and ViNet with our RealGuyot dataset, comprising annotated grapevine images acquired in real-world conditions, showing promising results for the use of EViNet in embedded devices.

However, there is still room for improvement, especially in the Courson area, which is critical for making pruning decisions. In addition, our RealGuyot dataset is limited in size; therefore, the acquisition of additional data would likely lead to improved performance. Finally, although our networks significantly improved runtime efficiency, additional optimization techniques such as quantization, pruning, or utilizing the ONNX format could further enhance efficiency.

7. Acknowledgements

The research was funded by the 3D2Cut company, the Idiap research institute, and the Fondation The Ark (Valais, Switzerland) through the projects 3D2cut and Vinet. The authors would also like to thank 3D2cut SA for sharing the RealGuyot dataset.

References

[1] Suchet Bargoti and James Underwood. Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3626–3633. IEEE, 2017. 2

[2] Tom Botterill, Scott Paulin, Richard Green, Samuel Williams, Jessica Lin, Valerie Saxton, Steven Mills, XiaoQi Chen, and Sam Corbett-Davies. A robot system for pruning grape vines. *Journal of Field Robotics*, 34(6):1100–1122, 2017. 1

[3] Emilie Bruez, Céline Cholet, Massimo Giudici, Marco Simonit, Tommasso Martignon, Mathilde Boisseau, Sandrine Weingartner, Xavier Poitou, Patrice Rey, and Laurence Geny-Denis. Pruning quality effects on desiccation cone installation and wood necrotization in three grapevine cultivars in france. *Horticulturae*, 8(8):681, 2022. 1

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 2

[5] A Casado-García, J Heras, A Milella, and R Marani. Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture. *Precision Agriculture*, pages 1–26, 2022. 2

[6] Sam Corbett-Davies, Tom Botterill, Richard Green, and Valerie Saxton. An expert system for automatically pruning vines. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, pages 55–60, 2012. 1

[7] Alain Deloire, Carole Dumont, Massimo Giudici, Suzy Rogiers, and Anne Pellegrino. A few words on grapevine winter buds and pruning in consideration of sap flow. *IVES Technical Reviews, vine and wine*, 2022. 1

[8] Gianni Fenu and Francesca Maridina Malloci. Forecasting plant and crop disease: an explorative study on current algorithms. *Big Data and Cognitive Computing*, 5(1):2, 2021. 2

Figure 5: Random set of images from the RealGuyot dataset processed by EViNet/R512U (second column) and ViNet (third column). Note that EViNet/R512U is six times faster than ViNet and 5 times lighter.

[9] Konstantinos P Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 145:311–318, 2018. 2

[10] Jaco Fourie, Christopher Bateman, Jeffrey Hsiao, Kapila Pahalawatta, Oliver Batchelor, Paul Epee Misse, and Armin Werner. Towards automated grape vine pruning: Learning by example using recurrent graph neural networks. *International Journal of Intelligent Systems*, 36(2):715–735, 2021. 1

[11] Theophile Gentilhomme, Michael Villamizar, Jerome Corre, and Jean-Marc Odobez. Towards smart pruning: Vinet, a deep-learning approach for grapevine structure estimation. *Computers and Electronics in Agriculture*, 207:107736, 2023. 1, 2, 3, 5, 6, 7

[12] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. 2

[13] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[15] Yaqoob Majeed, Manoj Karkee, and Qin Zhang. Estimating the trajectories of vine cordons in full foliage canopies for automated green shoot thinning in vineyards. *Computers and Electronics in Agriculture*, 176:105671, 2020. 2

[16] Yaqoob Majeed, Manoj Karkee, Qin Zhang, Longsheng Fu, and Matthew D. Whiting. Determining grapevine cordon shape for automated green shoot thinning using semantic segmentation-based deep learning networks. *Computers and Electronics in Agriculture*, 171:105308, 2020. 1

[17] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Investigating depth domain adaptation for efficient human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[18] Angel Martınez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Real-time convolutional networks for depth-based human pose estimation. in 2018 ieee. In *RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 2018, 2018. 2

[19] Angel Martínez-González, Michael Villamizar, and Jean-Marc Odobez. Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2276–2284, 2021. 2

[20] Zhonghua Miao, Xiaoyou Yu, Nan Li, Zhe Zhang, Chuangxin He, Zhao Li, Chunyu Deng, and Teng Sun. Efficient tomato harvesting robot based on image processing and deep learning. *Precision Agriculture*, pages 1–34, 2022. 2

[21] Annalisa Milella, Roberto Marani, Antonio Petitti, and Giulio Reina. In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Computers and electronics in agriculture*, 156:293–306, 2019. 2

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2, 3

[23] Fernando Palacios, Gloria Bueno, Jesús Salido, Maria P. Diago, Inés Hernández, and Javier Tardaguila. Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. *Computers and Electronics in Agriculture*, 178:105796, 2020. 2

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[25] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222, 2016. 2

[26] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468, 2019. 2

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2, 3, 4

[28] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[29] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint*, 2020. 2