COBEM 2021 26th International Congress of Mechanical Engineering
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-1076
NEURAL NETWORKS AND COMPUTATIONAL VISION APPLICATION IN FEATURE RECOGNITION AT AGRICULTURAL FIELDS

Lucas Toschi de Oliveira
Vitor Akihiro Hisano Higuti
University of São Paulo (USP), Av Trabalhador São-Carlense, 400, São Carlos, São Paulo, Brazil
ltoschi@usp.br, vitor.higuti@usp.br

Marcelo Becker
University of São Paulo (USP), Av Trabalhador São-Carlense, 400, São Carlos, São Paulo, Brazil
becker@sc.usp.br

Abstract. *The global population growth demands that the methods and technologies applied to food production must be even more efficient, making more with less space and resources. Furthermore, tasks like phenotype identification in large crops for research purposes are expensive and time-consuming. In the 1980s, precision agriculture arrived as a new concept, trying to power fundamental advances with technology. Nowadays, autonomous robotics is a trending research area to help with these problems due to its flexibility and portability. In this context, developing an independent navigation system capable of driving a robot inside a plantation is still challenging. However, it is the strategy used here, unlike most researchers that use a top view feature extraction. Among the navigation needs, local map creation with SLAM (Simultaneous Localization And Mapping) stands out and is a potential study subject. In dynamic places, like the agricultural one, this method is more susceptible to errors because of the typical assumption of static surroundings adopted in traditional approaches. In the search for better results, the use of Deep Learning in mobile object identification, which eliminates them from the mapping process, is a promising alternative and is the main focus of this study. Here, the algorithm focuses on the identification and classification of plants with boxes as a fundamental step towards SLAM application. Data utilized in the project was taken by TerraSentia, an agricultural mobile robot developed by researchers in LabRoM (Mobile Robotics Laboratory - EESC/USP) and Illinois University (Urbana-Champaign). This work uses the transfer learning method to train the lower layers of a convolutional neural network and develops a small dataset with CVAT (Computer Vision Annotation Tool) for its upper layers. The object detection algorithm is built in a ROS (Robot Operating System) node, allowing easy deployment in the robot's infrastructure and future projects. This research is part of a LabRoM set of analyses, which seeks a fully autonomous robot development for the agricultural environment. With the YOLOv3 implementation, the trained model was capable of plant detecting with a precision of 93.88% evaluated in 250 images from the validation set. Due to some dataset labeling imperfections, some discussions and new performance indicators were proposed.*

Keywords: *Agriculture, Computer Vision, Autonomous Navigation, Sensing, Deep Learning, Robotics, Mechatronics*

1. INTRODUCTION

In "An Essay on the Principle of Population" Thomas Malthus, an illuminist intellectual, defends that population growth has a geometric progression shape, whereas food production growth has arithmetic progression (i.e. slower than the first). According to the modern Malthusian theory, that is a difficulty that will never be overcome. However, following Revich *et al.* (2016), that is a rash conclusion, although the analysis is compatible with the global reality.

Revich *et al.* (2016) considers that humanity faces a food production capacity problem due to continuous population growth. In Fig. 1, it's possible to notice that until 2015, the global population showed a faster increase than the number of cultivable areas. Nonetheless, still according to the study, the answer to the future is in the technological advance, with development of autonomous vehicles and intelligent systems.

Concerning the necessity of more sophisticated tools to human life livelihood, precision agriculture arises as a new concept in the 1980s. This paradigm seeks food production quantity and quality growth while reducing costs. Robotics is a very versatile area, and it is reasonable in this context. However, most robotic navigation techniques were not developed in agricultural environments because they are more complex than urban ones (Gao *et al.*, 2018).

In the search for a completely autonomous agricultural system, mapping the local environment is indispensable. Therefore, Simultaneous Localization and Mapping (SLAM) is a technique that has great importance and still is a challenge in dynamic contexts, like the agricultural one.

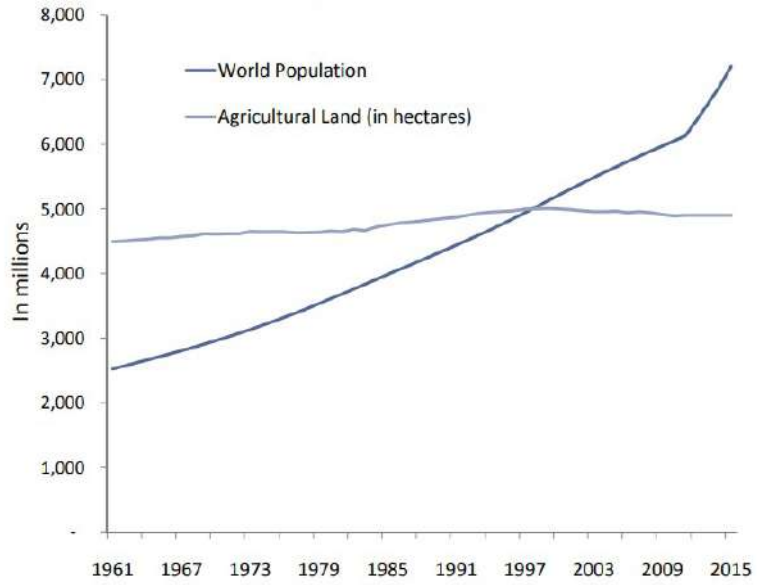

Figure 1. Global arable land acreage vs. population. Adapted from (Revich *et al.*, 2016)

The application of SLAM in these locations (also called Simultaneous Localization and Mapping in Dynamic Environments" - SLAMIDE) is difficult by the lack of an abstract understanding of the captured image. For this task, the use of artificial neural networks is attractive due to their precision in object detection and classification. Therefore, many researchers use them to find the mobile objects in a scene, ignoring them in the map construction.

The main objective of this project is the development of an artificial neural network able to locate characteristics in the agricultural environment. The neural network is expected to identify the upper portion of the plants in farming since they are the most mobile objects in the scene. It is believed that will enable more efficient implementations of SLAM in the future. For this research, the images were taken by TerraSentia, an agricultural robot equipped with a monocular camera. Therefore, the main contribution of this work is the study of object detection with data obtained in an under-canopy scenario. The plants' movement in the agricultural environment difficulties the use of computational vision traditional methods, requiring Deep Learning implementations.

In this paper, some topics will be addressed. The first one is the mobile platform description, which briefly describes the TerraSentia robot. Next, a literature review will be presented, showing some Deep Learning models used along with SLAM algorithms. After that, the research methodology will be described including points like dataset production, computational power, and mathematical basis. Finally, the YOLOv3 implementation results and the model validation will be discussed.

2. MOBILE PLATFORM DESCRIPTION

TerraSentia (shown in Fig. 2) is a low-cost, ultracompact (0.31m wide), and ultralight (6.6 kg) 3D-printed field robot developed by University of Illinois at Urbana-Champaign and provided to LabRoM. It was described by Kayacan *et al.* (2018), Higuti *et al.* (2018) and it is designed to under-canopy movement. The platform has a Raspberry Pi 3 B minicomputer and can be equipped with different sensors (like cameras and LiDAR). For this project, the images were obtained by a 2.0 USB 5Mp OMNIVISION OV5640 color CMOS sensor 2,1MM perspective camera.

3. LITERATURE REVIEW

3.1 SLAM and Deep Learning

The usual SLAM methods are constructed based on the static environment assumption, where there are not objects moving in the scene. This assumption limits the development of this technology for real-life applications, whereas the unstable points capture results in large trajectory errors and even complete navigation system failure (Han and Xi, 2020). SLAM application in dynamic enviroments is called SLAMIDE and it is a relevant research topic in navigation robot studies (Xiao *et al.*, 2019).

Using SLAM with image data (called Visual SLAM) is based on four basic steps: tracking, mapping, global optimization, and relocation. The first one consists of using consecutive images in the local trajectory generation as well as in obtaining depth info; the second refers to the virtual map generation process with sensor data. The third is the global mapping correction, removing slipping errors. The last concerns obtaining the robot's localization from the virtual map

Figure 2. TerraSentia platform and its camera. Adapted from EarthSense (2021) and ELPCCTV (2020)

when in an unknown position (Milz *et al.*, 2018).

Deep Learning-based implementations are considered an excellent solution to the SLAMIDE challenge due to their good performance in tasks related to data association - although they perform not so well in scenarios not included in the training. Therefore, researchers focus their work on complementing subsystems instead of trying to develop a full process model (Kang *et al.*, 2019).

There are several good opportunities for Deep Learning technologies like depth estimation, optical flux identification (image frames displacement), characteristics correspondence, and semantic segmentation (Milz *et al.*, 2018). In particular, semantic understanding of the scene (the focus of this work) is used in object point extraction processing and it is fundamental in high-level task planning (Xiao *et al.*, 2019) (Yu *et al.*, 2018).

3.2 Semantic Segmentation

Semantic segmentation consists of image subdivision in relevant semantic regions, classifying them according to predetermined categories. There are three main subcategories: training based on packages, dense classification, and multi-scale classification. In the first, high-resolution images are fragmented and are used during training; in the second, convolutions (and de-convolutions) are utilized to classify all pixels in the scene (as in Yu *et al.* (2018)). In the last one, different image resolutions are considered during the network's work (Milz *et al.*, 2018).

For multi-scale classification, the neural networks that stands out are those capable of doing object detection and classification with only one forward pass through the model. Besides having good accuracy, they are sufficiently efficient in almost real-time execution. Some of these methods will be presented next.

3.2.1 Dynamic-SLAM

The first method presented is called Dynamic-SLAM (Xiao *et al.*, 2019), which uses the Single Shot Detector (SSD) neural network (Fig. 3 is the model representation) - developed by Liu *et al.* (2016). It consists of two elementary operations: convolution and pooling.

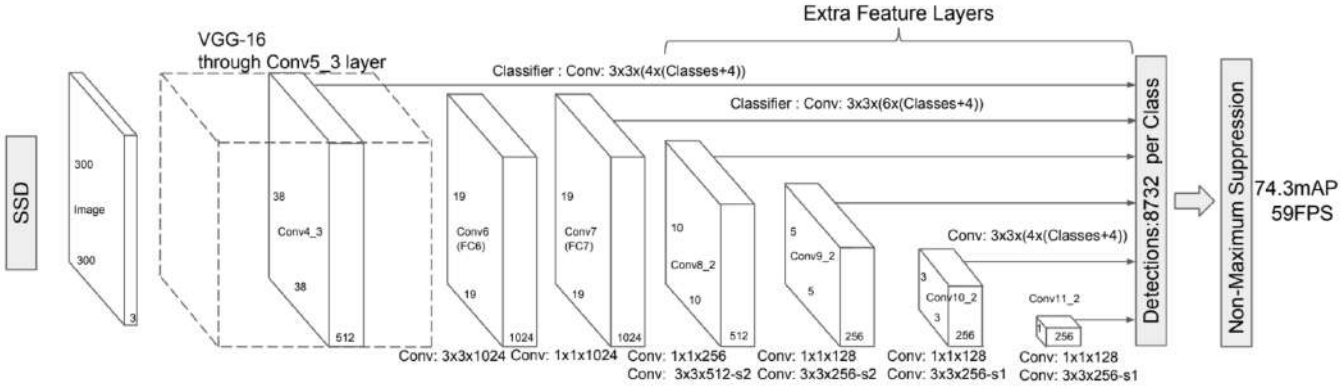

Figure 3. Single Shot Detection (SSD) model. Image from Liu *et al.* (2016)

The first layers of the SSD are part of a pre-trained neural network called VGG-16 (Simonyan and Zisserman, 2015).

Above it, other layers are stacked and their results organized as the final network predictions - this structure allows the detection of different size features. The standard boxes shapes are manually established from the objects expected dimensions.

Convolutionally, the neural network relates the objects from their center position concerning each image cell. For each cell from a classification layer with size m x n, the k standard boxes have scores for the c predetermined classes and four values that represent the box offset with respect to the quadrant center. Therefore, for each image cell, (c + 4)kmn outputs are generated.

When neural networks perform localization and classification tasks, both are considered to describe the total performance. For localization, the generated boxes are compared with those from the dataset using the IoU metric (also called Jacard's index). For classification, the softmax layer result for each default box is the probability that the determined object be part of each class.

For one image, the method described above performs many predictions for classification and localization. The Non-Maximum Suppression method is executed to obtain the best one as the final output for the network - only the best boxes according to the metrics already mentioned remain, and the others are discarded.

3.2.2 Semantic SLAM

The second method is called Semantic SLAM. With this name, two different articles were found. Although they presented different strategies for SLAM application, they use similar techniques for semantic segmentation. Thus Zhang *et al.* (2018) utilizes the YOLO neural network, whereas Qian *et al.* (2020) uses its evolution, YOLOv3.

You Only Look Once (YOLO) is a famous neural network capable of performing localization and classification with only one forward pass (Redmon *et al.*, 2016). The image is divided convolutionally, and the objects are identified from the default box definition. The model is represented in Fig. 4.

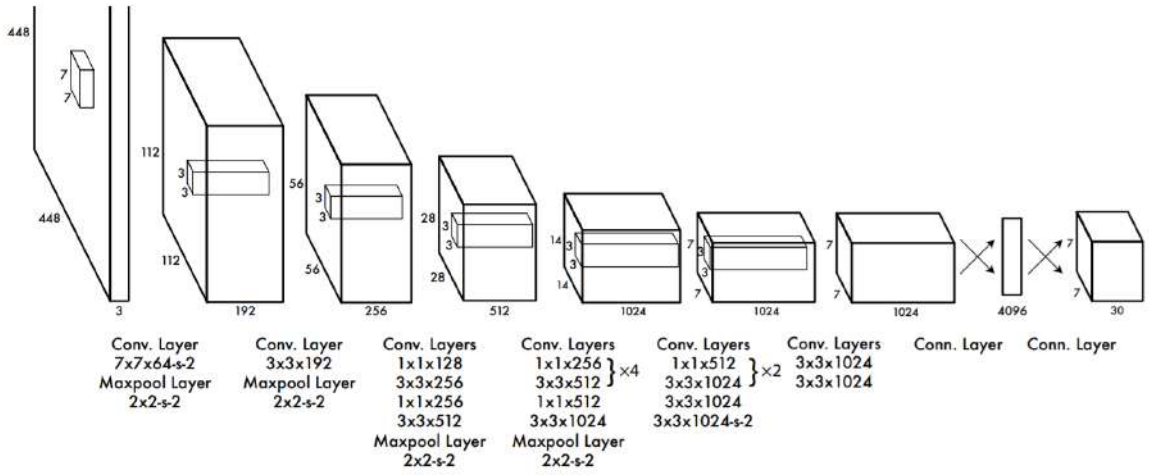

Figure 4. You Only Look Once (YOLO) model. Image from Redmon *et al.* (2016)

Its evolution, YOLOv3, can perform multi-scale object detection (like SSD mentioned before), connecting some previous layers to its output (Redmon and Farhadi, 2018). As shown in Fig. 4, she is based on the Darknet53 network, whose main function is to do convolutions and skip connections extracting basic image features. After different size layers are concatenated (19x19, 38x38, 76x76), the network output consists of box offsets and sizes. Besides that, each box has scores: the object existence probability and each class probability.

3.2.3 PSPNet-SLAM

The third method is called PSPNet-SLAM (Han and Xi, 2020). It uses the PSPNet network (Zhao *et al.*, 2017) to perform semantic segmentation, classifying their pixels. The model is represented in Fig. 6. According to the authors, this method is more efficient than a Fully Connected Network (FNC).

The PSPNet network is based on Pyramid Pooling Module, a predetermined layer set. Initially, the entry image is passed by a convolutional neural network for helpful feature extraction (it is used the ResNet by He *et al.* (2016)). After that, the pooling operation is realized in different patterns, generating different image portions that go through convolutions to feature extraction. This structure allows the image interpretation in various contexts, obtaining better results.

After that, each layer passes by an upsampling process (through bilinear interpolation) which preserves the input

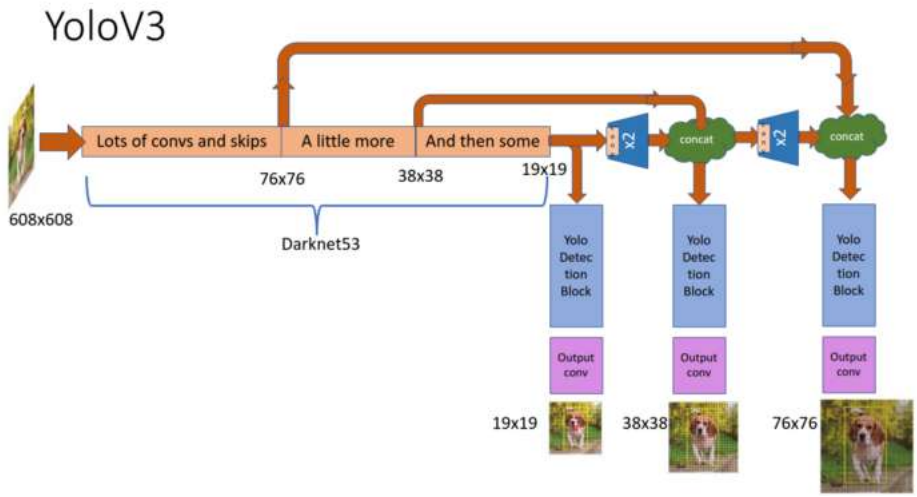

Figure 5. You Only Look Once v3 (YOLOv3) model. Image from Redmon and Farhadi (2018)

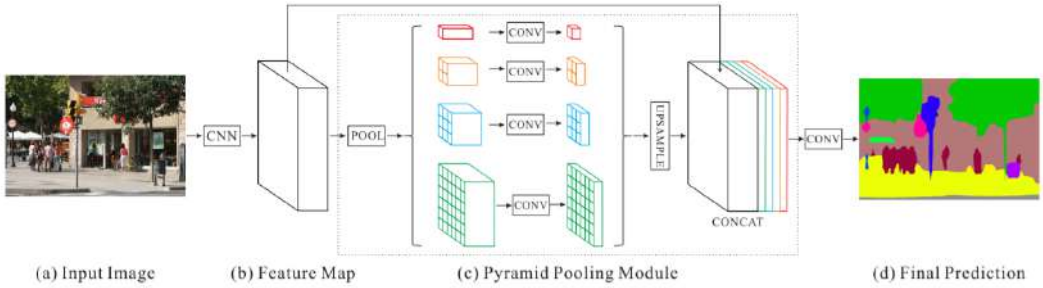

Figure 6. Pyramid Scene Parsing Network model. Image from Zhao *et al.* (2017)

image dimensions in the neural network's output. According to the authors, the pyramidal module can be changed (size and shape) to obtain different results. In the original implementation, the layers used were 1x1, 2x2, 3x3, and 6x6.

4. METHODOLOGY

In Deep Learning algorithms implementation, frameworks are often used. The majority of them are available in the market for free. In this project, the framework Keras was used, built around TensorFlow, a Python library from Google.

Among the neural networks described in the Bibliography Review section, the YOLOv3 model was the first one tested. It was chosen due to the available free-to-use implementations made by the Deep Learning community. The Balsys (2019) implementation on GitHub was used as a guide, including techniques like image augmentation and batch-size training. Besides that, transfer learning for the backbone network (Darknet53), an input image shape of 416x416 pixels and the Adam global optimizer were also adopted.

4.1 Training process

In the neural network training process, some items are required: a global optimizer algorithm, computational power and a labeled dataset.

4.1.1 Global optimizer

The learning algorithm used in this project includes a weights global optimizer, called Adam (Kingma and Ba, 2017). To facilitate the explanation, supposes a Fully Connected Network (FNC) which every output of every neuron from a layer is connected to another neuron in the next layer. Therefore, the linear function describing the neuron activation is

$$Z^{[L]}_{n_L \times m} = W^{[L]}_{n_L \times n_{(L-1)}} A^{[L-1]}_{n_{(L-1)} \times m} + B^{[L]}_{n_L \times 1} \tag{1}$$

In Eq. 1, Z is the activation matrix (with shape $n_L \times m$, where n_L is the number of neurons in the layer L and m is the number of images used in the batch); W is the multiplicative weight matrix (with shape of $n_L \times n_{(L-1)}$, where $n_{(L-1)}$ is the number of neurons in the previous layer); A is the non-linear activation matrix (with shape $n_{(L-1)} \times m$); B is the

additive weight matrix (with shape $n_{(L-1)} \times m$). In this notation, the last sum (B matrix) is done in every single column from the W and A multiplication result.

For a certain layer, the A matrix is obtained by using a non-linear function in the Z matrix. This allows the neural network to learn complicated patterns and effectively be useful to real-world problems. One of the most used functions is the ReLU activation function that is shown in Eq. 2 (z is each element of the Z matrix).

$$g(z) = max(0, z) \tag{2}$$

The learning process is, in fact, a weight tuning problem. In Adam global optimization, the Eqs. 3 and 4 are the expressions that define the weight change for each iteration i.

$$w_p^i = w_p^{i-1} - \alpha \frac{V_{dw}}{\sqrt{S_{dw}}} \frac{\sqrt{(1-\beta_2^t)}}{(1-\beta_1^t)} \tag{3}$$

$$b_p^i = b_p^{i-1} - \alpha \frac{V_{db}}{\sqrt{S_{db}}} \frac{\sqrt{(1-\beta_2^t)}}{(1-\beta_1^t)} \tag{4}$$

In both, α, β_1 and β_2 are called hyperparameters, manually tuned before training. V_{dw}, V_{db}, S_{dw} and S_{db} are auxiliar variables, defined in the Eqs. 5, 6, 7 and 8.

$$V_{dw} = \beta_1 V_{dw} + (1-\beta_1) d_w \tag{5}$$

$$V_{db} = \beta_1 V_{db} + (1-\beta_1) d_b \tag{6}$$

$$S_{dw} = \beta_2 S_{dw} + (1-\beta_2) d_w^2 \tag{7}$$

$$S_{db} = \beta_2 S_{db} + (1-\beta_2) d_b^2 \tag{8}$$

In the four equations above, d_w is the derivative of the error function (that depends on the task required) in relation to the w parameter and d_b is the derivative of the error function relative to each b weight.

4.1.2 Computational power

This research is being conducted in LabRoM, the Robotic Mobile Laboratory from the University of São Paulo (USP, Brazil). This laboratory uses the infrastructure provided by Amazon Web Services (AWS), including on-demand computing and storage. For the initial tests, it was used a "p2.xlarge" machine instance. It has 1 GPU with 12 GiB of memory and 4 CPUs with 61 GiB of memory. The GPU is a high-performance NVIDIA K80, with 2496 processing cores (Services, 2021).

4.1.3 Labeled dataset

Using TerraSentia images captured in real-life agriculture fields, a new dataset was constructed with the Computer Vision Annotation Tool (CVAT), shown in Fig. 7.

Possible to be utilized in a local or online environment, the CVAT allows the development of different tasks inside the same project. Besides that, the tool has a box tracking option, facilitating object annotation along with various frames. When creating a box, it is interpolated in the next frames requiring just small adjustments to label an image. CVAT can export the labels in various formats, in particular in the same format that the dataset PASCAL VOC, more convenient with the found implementations.

5. EXPECTED AND INITIAL RESULTS

As previously stated, the main objective of this project is to develop a neural network capable of identifying the plant's middle section. The expected final results in the validation dataset sample are shown in Fig. 8 by the pink boxes. In this figure, it is possible to understand what is meant by "middle section": it refers to the region above the plant's root, near the main stem.

The first accomplishment was the dataset construction, already with one thousand images. Also in Fig. 8, it is possible to check a sample of it, generated from the images taken by TerraSentia from a cornfield. The last row examples are problematic ones since the camera is partially obstructed by plants' leaves. It is expected that the neural network would be robust even in these cases, identifying the plants that are not hidden.

The dataset was divided into two parts: one for training (750 images) and another for validation (250 images). All of them are images taken from cornfields, but it will be expanded in size and variety as the project advances.

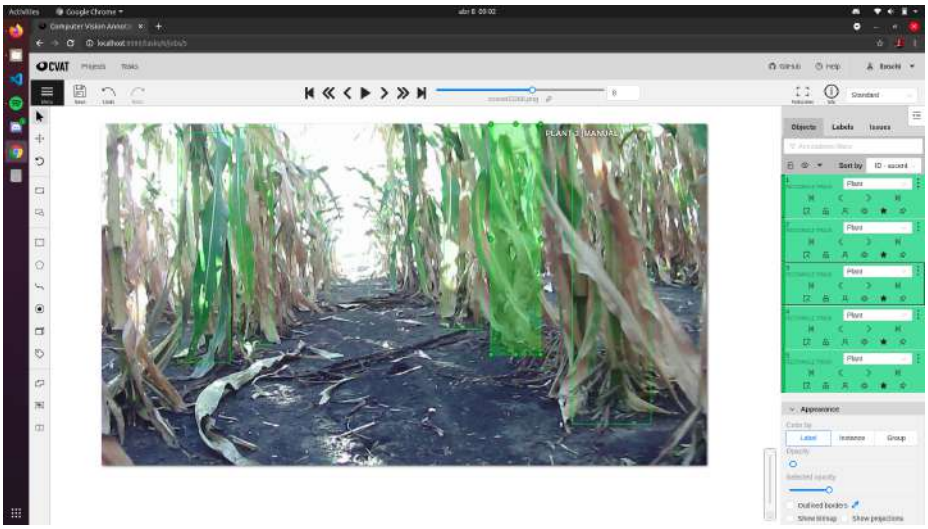

Figure 7. Computer Vision Annotation Tool (CVAT) interface

Figure 8. Expected neural network predictions in validation sample

5.1 YOLOv3 results

Figure 9 describes the training process of the YOLOv3 model - the left plot is the validation loss and the right one is the training loss. To complete the training, 87 steps were needed until the early stopping activation, that is, when the validation loss does not improve over training generations.

Figure 10 shows some images obtained applying the trained YOLOv3 model in the validation dataset. The pink boxes represent the ground-truth boxes and the yellow boxes represent the model predictions. The latter is accompanied by the label "Plant" and a number, indicating the image classification outcome and the confidence level of the model results.

Some observations can be made: in plots (a) and (b), the neural network found good results that were not manually marked in the ground-truth boxes; conversely, in plots (c) and (d), the neural network failed to detect some of the ground-truth boxes.

The first finding can be explained by the ground-truth boxes' nature: they were manually selected by a human. In other words, the boxes were subjectively chosen, sometimes considering more evident plants and other times including more hidden ones. Therefore, the existence of imprecisions in the box placement criteria was, in fact, a natural consequence of the dataset labeling method. Surprisingly, the neural network was capable of abstracting the plants' pixel information and generate good results even with missing ground boxes in the training set.

On the other hand, the second observation can be explained by the neural network's nature. It fails to detect all the

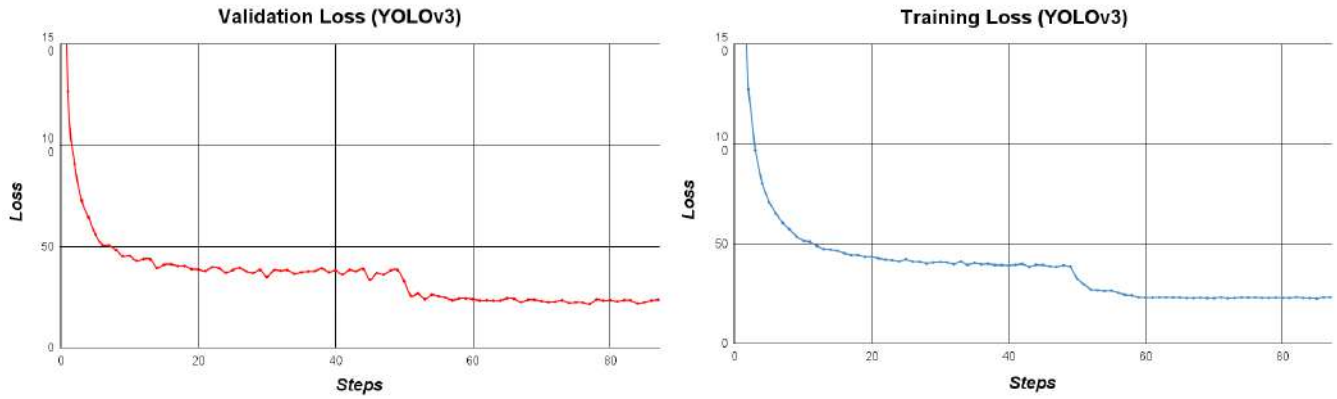

Figure 9. YOLOv3 validation loss (left) and model training (right)

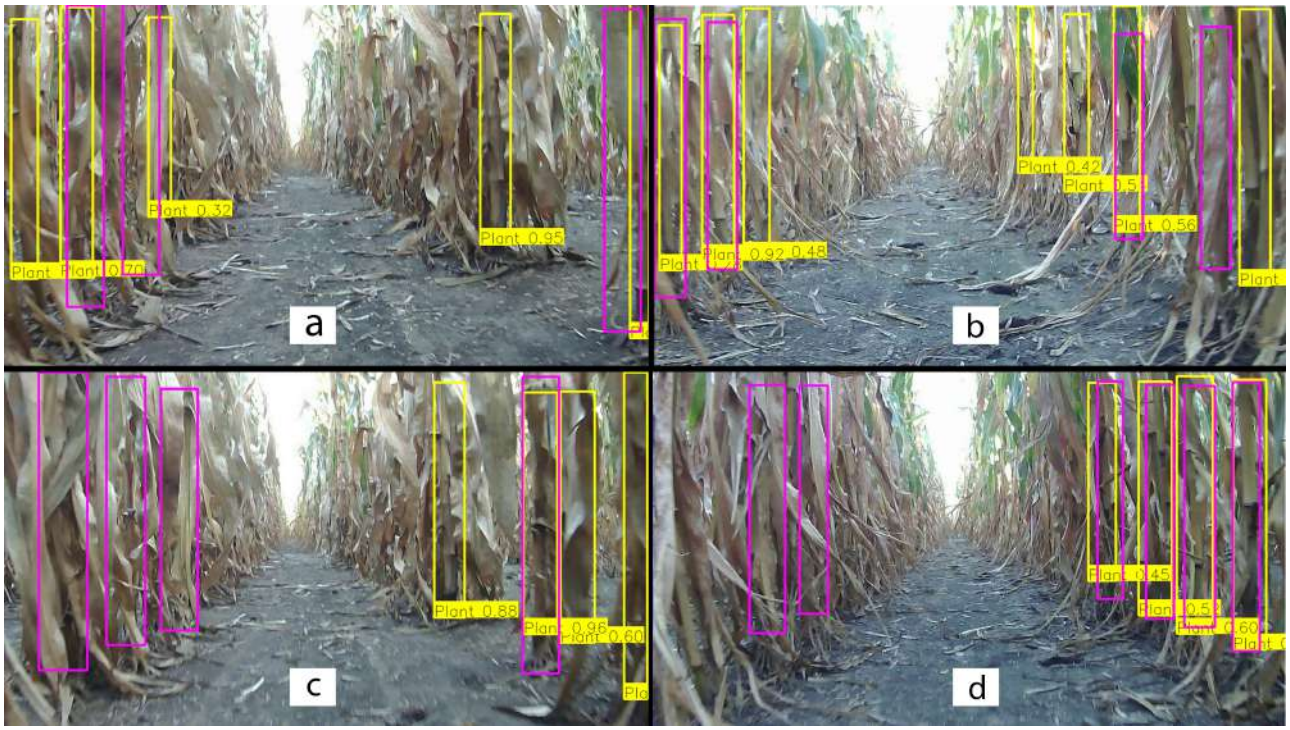

Figure 10. Some YOLOv3 model predictions (in yellow) and ground-truth boxes (in pink)

plants in the scenario, especially the ones that are more occluded by leaves. For future improvement, dataset expansion and review can be done to obtain even better results.

Usually, in object detection, the Mean Average Precision (mAP) is used to measure how much the model accomplished to adjust the predictions to the ground boxes. The calculated mAP value is 65.70%, a result that is not good to understand the neural network performance since some identifiable plants were omitted in labeling as mentioned previously.

Therefore, to measure the model precision, custom metrics were necessary. In each of the 250 images from the validation dataset, the number of ground boxes (n_{gt}) and the number of predictions (n_{pred}) were measured. Besides that, it was counted the number of good predictions boxes (n_{gpred}) and the amount of them that coincide with the ground truth ones (n_{gtpred}). The term "good predictions" was oriented by the question: *"does this box delimit the majority of a plant's pixels?"*.

With the described metrics, it was possible to calculate the virtual "total" of plants that were identifiable in each frame (n_{vtotal}) given by the Eq. 9, considering the human criteria and the neural network generalization.

$$n_{vtotal} = n_{gt} + (n_{gpred} - n_{gtpred}) \tag{9}$$

The first parameter to evaluate the neural network performance is the precision (P). That can be calculated from the n_{gpred} and the n_{pred} variables, according to Eq. 10.

$$P = \frac{n_{gpred}}{n_{pred}} \tag{10}$$

The second one measures the coincidence (C) of the predictions and the ground truth boxes in relation of the total of yellow boxes. This relationship is described in Eq. 11.

$$C = \frac{n_{gtpred}}{n_{pred}} \tag{11}$$

The third one (NC) measures how many predictions boxes do not coincide with the ground truth ones in relation to the total of the ground truth boxes (Eq. 12).

$$NC = \frac{n_{gt} - n_{gtpred}}{n_{gt}} \tag{12}$$

The fourth one (EB) measures how many good predictions that not coincide with the ground truth boxes were produced in relation to the total of predictions (Eq. 13).

$$EB = \frac{n_{gpred} - n_{gtpred}}{n_{pred}} \tag{13}$$

The fifth one (EBOT) measures how many predictions that not coincide with the ground truth boxes were produced in relation to the virtual "total" of plants (Eq. 14).

$$EBOT = \frac{n_{gpred} - n_{gtpred}}{n_{vtotal}} \tag{14}$$

Calculating the average of the five parameters for each image from the validation dataset, the Table 1 was constructed (prefix "m" indicates the mean value).

Table 1. Mean YOLOv3 model performance parameters values

mAP	**mP**	**mC**	**mNC**	**mEB**	**mEBOT**
65.70%	93.88%	62.61%	25.71%	31.48%	26.18%

Observing Table 1, it is possible to see that the mean overall precision for the model (mP) is relatively good (93.88%). Still only 62.61% (mC) of the predictions coincide with the ground truth boxes, 31.48% (mEB) of the predictions were good and do not coincide with ground truth boxes. Concerning the virtual "total" of plants in each image, 26.18% of the predictions do not have some previous reference in the dataset. Nonetheless, 25.71% (mNC) of the ground truth boxes were not identified by the model.

6. CONCLUSIONS

Due to the accelerated population growth, robotics is a constantly growing area to increase food production. Therefore, this work proposes the application of deep learning in the agricultural context aiming to improve robotic navigation methods. The main objective is to identify the plants in the scenario with boxes for the posterior application of SLAM algorithms. The neural network was trained with images captured by TerraSentia, a robot designed to under canopy movement. The performance analysis indicates that the YOLOv3 model was successful in the object detection task. Even if some ground truth boxes of the validation dataset were not identified, the algorithm generalization works well enough to detect the plants ignored in the manual labeling.

To achieve even better results in the future, the next step involves dataset diversity expansion, considering more culture types. Furthermore, it can be enlarged in size, providing more data for neural network generalization. Besides that, to decrease subjectivity, the manual labeling process can include more people defining where the ground truth boxes must be in each image. Finally, other models can be tested to check if they achieve better results for plants' detection in the agricultural context.

7. ACKNOWLEDGEMENTS

This paper was supported by grants no. 2020/11262-0 and no. 2018/10894-2, São Paulo Research Foundation (FAPESP)

8. REFERENCES

Balsys, R., 2019. "Training custom yolo v3 object detector". URL `https://pylessons.com/YOLOv3-custom-training/`. Acessado em 9 aug. 2021.

EarthSense, I., 2021. "Earthsense agricultural intelligence". URL `https://www.earthsense.co/`. Acessado em 13 jun. 2021.

ELPCCTV, 2020. "5megapixel usb camera module usb2.0 omnivision ov5640 color cmos sensor 2.1mm lens". URL `http://www.elpcctv.com/5megapixel-usb-camera-module-usb20-omnivision-ov5640-color-cmos-sensor-21mm-le` Acessado em 18 ago. 2020.

Gao, X., Li, J., Fan, L., Zhou, Q., Yin, K., Wang, J., Song, C., Huang, L. and Wang, Z., 2018. "Review of wheeled mobile robots' navigation problems and application prospects in agriculture". *IEEE Access*, Vol. 6, pp. 49248–49268.

Han, S. and Xi, Z., 2020. "Dynamic scene semantics slam based on semantic segmentation". *IEEE Access*, Vol. 8, pp. 43563–43570.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. "Deep residual learning for image recognition". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778.

Higuti, V., Velasquez, A., Magalhães, D., Becker, M. and Chowdhary, G., 2018. "Under canopy light detection and ranging-based autonomous navigation". *Journal of Field Robotics*, Vol. 36. doi:10.1002/rob.21852.

Kang, R., Shi, J., Li, X., Liu, Y. and Liu, X., 2019. "Df-slam: A deep-learning enhanced visual slam system based on deep local features".

Kayacan, E., Zhang, Z. and Chowdhary, G., 2018. "Embedded high precision control and corn stand counting algorithms for an ultra-compact 3d printed field robot". doi:10.15607/RSS.2018.XIV.036.

Kingma, D.P. and Ba, J., 2017. "Adam: A method for stochastic optimization".

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. "Ssd: Single shot multibox detector". *Lecture Notes in Computer Science*, p. 21–37. ISSN 1611-3349.

Milz, S., Arbeiter, G., Witt, C., Abdallah, B. and Yogamani, S., 2018. "Visual slam for automated driving: Exploring the applications of deep learning". In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. pp. 360–36010.

Qian, Z., Patath, K., Fu, J. and Xiao, J., 2020. "Semantic slam with autonomous object-level data association".

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. "You only look once: Unified, real-time object detection". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 779–788.

Redmon, J. and Farhadi, A., 2018. "Yolov3: An incremental improvement".

Revich, J., Boroujerdi, R.D., Scott-Gall, H., Patrick Archambault CFA, R.K.C., Samuelson, A., Nannizzi, M., CFA, S.G., Moawalla, M., Hulsing, J., CFA, N.P., Burgstaller, S., Isayama, Y., Bonin, A., CFA, D.T., Yang, J., Roach, B., Porat, M., CFA, L.K., Agarwal, D., Pillai, G., Ph.D, C.E., Berney, R., Cohen, D. and Verma, G., 2016. "Precision Farming - Cheating Malthus with Digital Agriculture".

Services, A.W., 2021. "Amazon ec2 instance types". URL `https://aws.amazon.com/en/ec2/instance-types/`. Acessado em 13 jun. 2021.

Simonyan, K. and Zisserman, A., 2015. "Very deep convolutional networks for large-scale image recognition".

Xiao, L., Wang, J., Qiu, X., Rong, Z. and Zou, X., 2019. "Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment". *Robotics and Autonomous Systems*, Vol. 117, pp. 1 – 16. ISSN 0921-8890.

Yu, C., Liu, Z., Liu, X., Xie, F., Yang, Y., Wei, Q. and Fei, Q., 2018. "Ds-slam: A semantic visual slam towards dynamic environments". In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1168–1174.

Zhang, L., Wei, L., Shen, P., Wei, W., Zhu, G. and Song, J., 2018. "Semantic slam based on object detection and improved octomap". *IEEE Access*, Vol. 6, pp. 75545–75559.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J., 2017. "Pyramid scene parsing network".

9. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.