Deep Learning Algorithm for Object Detection with Depth Measurement in Precision Agriculture

Aguirre Santiago, Leonardo Solaque and Alexandra Velasco
Department of Engineering, Universidad Militar Nueva Granada, Bogot, Colombia

Keywords: GPU, Object Detection, Deep Learning, Depth Measurement, Point Cloud, Agricultural Robot.

Abstract: Autonomous driving in precision agriculture will have an important impact for the field. This is why several efforts have been done in this direction. We have developed an agricultural robotic platform named CERES, which has a payload of 100 Kg of solid fertilizer, 20 liters for fumigating purposes, and a weeding system. Our research points to make this robot autonomous. In this paper, we propose a method, based on deep learning algorithms, to combine object detection with depth measurements for object tracking and decision making of an agro-robot. For this, we combine an object detection algorithm carried out with YOLOv2 and a depth measurement strategy implemented with a ZED Camera. The main purpose is to determine the distance to the obstacles, mainly people, because we require to prevent collisions and damages either for people and for the robot. We have chosen to detect people because, in the desired environment, these are frequent and unpredictable obstacles, and the risk of collision may be high.We use a host computer, achieving a detection network with an average accuracy of up to 72% in detecting the class Person. While using a NVIDIA Jetson TX1, the accuracy increases up to 84% due to the powerful dedicated GPU destined to process Convolutional Neural Networks(CNN).

1 INTRODUCTION

Machine learning has recently gained much attention due to several possible applications such as (Chlingaryan et al., 2018),(Shin et al., 2020), and (Espejo-Garcia et al., 2018). One of these applications is computer vision. In this field, object detection is useful in areas of study as medicine (e.g. (Li et al., 2019), (Zhou et al., 2019) (Chua et al., 2019)), autonomous driving (e.g. (Fujiyoshi et al., 2019), (Chen et al., 2018)), and precision agriculture (e.g. (Patrcio and Rieder, 2018), (Partel et al., 2019)), among others.

Several machine learning techniques for object detection have been already developed with good results. For example, a method based on deep convolution neural networks, released in 2014 is the Region-based Convolutional Network (R-CNN) (Wu et al., 2020). Since then, there have been improvements to this technique, e.g. Fast R-CNN (Girshick, 2015), and Faster R-CNN. (Ren et al., 2015). Other detection networks, such as YOLO (YouOnlyLookOnce) (Redmon and Farhadi, 2017) can be also used with similar purposes, i.e. object detection. For further information on this topic, the reader is encouraged to review (Wu et al., 2020).

On the other hand, depth measurement is an extra variable that can be obtained by different methods and using environments such as the ones presented in (Silva et al., 2020), (Kopp et al., 2019), and (Breton

Figure 1: CERES robot: Agricultural robotic platform electrically powered with two brushless motors liquid-cooled (each motor is 5KW) attached to a gearbox 50:1 these are coupled to a car wheel of a common rin 14. CERES has a payload of 100 Kg of solid fertilizer, 20 liters for fumigating purposes and a weeding system. For the high-level processes, CERES uses a 9 DOF IMU, a LIDAR and stereo cameras are integrated.

Santiago, A., Solaque, L. and Velasco, A.
Deep Learning Algorithm for Object Detection with Depth Measurement in Precision Agriculture.
DOI: 10.5220/0009869404900497
In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2020)*, pages 490-497
ISBN: 978-989-758-442-8

et al., 2019), just to name a few.

In fields like autonomous driving, to make decisions during a trajectory execution, the system requires obstacle detection. From obstacle detection, it may be possible to obtain other information like depth, which is useful for recalculating the trajectory. In this work, we tackle the topic of autonomous driving in precision agriculture. We are particularly interested in object detection with depth measurement for obstacle avoidance and decision making. Therefore, we propose a method to combine an object detection algorithm with depth measurement at the same time. This method will be applied to the CERES Agrobot, shown in Fig. 1. The idea is to use both obstacle detection and depth measurements in the robot navigation for decision making to prevent collisions and damages to the robot and to people that could interact somehow with the robot.

In the literature, there are several solutions to the problem of object detection, and in some cases, also depth measurement is presented, with different targets. For example, regarding the autonomous robot for agriculture in (Sadgrove et al., 2018), authors propose a cascading algorithm for fast feature extraction and object classification, resulting in an object detection algorithm. Moreover, related to Intelligent Transportation Systems, in (Hendry and Chen, 2019) an Automatic License Plate Recognition is presented, which consists of four steps, i.e. image collection, object detection, segmentation, and optical character recognition. For this application, authors use a reduced version of the first release of YOLO network (Redmon and Farhadi, 2017). However, to increase the precision accuracy a filter is required. A further application of object detection algorithms is the human action recognition to detect motion proposed by (Shinde et al., 2018), where the authors use the first release of YOLO to analyze human actions, but authors do not provide depth measurement. Regarding depth measurement, there are several techniques that allow good results (see e.g. (Zhao et al., 2017), (Li et al., 2018), and (Reiss et al., 2014)). Depth information can be extracted, for example, from two-dimensional data as in (Ban and Lee, 2020), where authors propose a method for obtaining important features of a depth image analyzing inherent feature that represents three-dimensional protuberance by using only two-dimensional distance information estimating details of a scene as a visual detection application.

In some fields like agriculture, it is necessary to have either the information of object detection and depth measurements for applications like autonomous driving. To make decisions, we need to know whether there is an obstacle, and its distance to the robot, e.g. to avoid the obstacle and to re-plan a trajectory. One way to solve both problems at the same time, either obstacle detection, and depth measurement is by position tracking. For instance, in (Hu et al., 2018) a theoretical control scheme for robust position tracking of a helicopter is proposed, but it needs to be tested on an experimental system.

Object detection and depth measurement may allow to solve similar problems. For example, the object tracking problem in one hand can be achieved by analyzing the depth, measured by a LiDAR sensor (Gong et al., 2020). On the other hand, the same problem can be solved by training a detection Network such as YOLO and analyzing the changes in the generated detection (Ciaparrone et al., 2020).

In this paper, we propose a method to combine object detection with depth measurements for object tracking and decision making of an agro-robot. For this, we combine an object detection algorithm carried out with YOLOv2 and a depth measurement strategy implemented with a ZED Camera. Fig. 2 illustrates the implemented strategy using the ZED SDK camera for image acquisition, followed by the image processing carried out using OpenCV library (Bradski, 2000). Then, with Matlab, we generate a static library with the GPU coder, to build the whole algorithm on ROS (Robot Operating System), and embedding the solution in a quad-core ARM Cortex-A57, 4GB LPDDR4 and integrated 256-core Maxwell GPU, Nvidia Jetson Tx1 module. The host computer is an Intel Core i5-7200 with 2 GB NVIDIA Geforce MX940 GPU. The main idea of our work is to determine the distance to the obstacles, mainly people, because we require to preserve either persons' and our CERES agricultural robot's integrity, for which we use a deep learning strategy combined with depth measurement to re-plan the trajectory. We have chosen to detect people because, in the desired environment, these are frequent and unpredictable obstacles, and the risk of collision is high. Here we address the strategy to detect the object and its distance to the robot, while the trajectory planning is not part of this paper. Using the host computer, we achieve a detection network with an average accuracy of up to 72% detecting the class "Person". The depth measurements not acquired with this host computer. Moreover, using a NVIDIA Jetson Tx1 supercomputer module, we obtained an accuracy of up to 84% detecting the the class Person. In this case, regarding depth measurement, we can detect objects in a range from 0.5 m up to 8m with an error around 3%, which gives us the capacity of re-planning the trajectory.

In section 2 we present the complete strategy for object detection using depth measurements as well.

The people detector training process, using YOLOv2 network, is fully described. Section 3 presents the implementation of our strategy on the NVIDIA Jetson Supercomputer. We analyze the results in section 4, and we give some conclusions and recommendations in section 5.

2 OBJECT DETECTION STRATEGY COMBINED WITH DEPTH MEASUREMENT

In this section, we describe the strategy used to detect an object and determine its distance to the robot. To do this, we trained the YOLOv2 detection network (Redmon and Farhadi, 2017) and then, we are able to define the distance between the camera and the object. The used network works with a single neural network applied to the full image, which divides the image into regions and predicts bounding boxes. The latter is used to perform depth measurement tasks. The architecture of the YOLOv2 network consists of 24 layers. The input layer uses a RGB image, while the output layer has 4 anchors, for more information about this detection network the reader can refer to (Redmon and Farhadi, 2017)[1].

For the purpose of our work, we trained the network to detect people using the INRIA person dataset (Taiana et al., 2013) and the PennFudanPed dataset (Ciaparrone et al., 2020). The former dataset contains a train set with 614 positive images, while the test set has 288 positive images (Ding and Xiao, 2012). The latter dataset contains a train set of 170 positive images. Both datasets have a complex background with a remarkable light change. Several features are considered, so both datasets are very useful for the purposes of this work.

2.1 People Detector Training Process

People detection algorithms have many applications such as in autonomous driving. For example, companies like Tesla, Apple, Toyota, Nissan, etc., use them to avoid collisions during a course (Wang et al., 2020). As mentioned before, we are interested in avoiding people to preserve their integrity as well as the integrity of CERES agricultural robot. For this, we detect the class person by training the YoloV2 detection network. The process followed is illustrated in Fig. 3. The first step includes the selection of a convolutional neural network (CNN) to edit the architecture and rebuild it as a Yolov2 sub-detection Network. Then, the second step is to select a strong dataset to train the detection network. In this case, as we already explained, we used two datasets which together contain 884 images. Finally, the third step is to set the training options based on the capacity of the host Computer, which in this case is the Intel Corei5-7200 with 2GB NVIDIA Geforce MX940 GPU. Then, we label all the images using the Image Labeler application provided by Matlab. Finally, we can train and test the detection network.

2.2 Depth Acquisition

In this paper the algorithms are developed using a supercomputer NVIDIA Jetson TX1 with Jetpack 3.3 OS, as well as ROS and the ZED SDK (Developer kit). To measure the distance to the obstacle, we chose a ZED stereo camera which uses triangulation and 3D sensors to estimate the depth from the disparity image. Fig. 4, presents the flowchart of the strategy built on ROS. To combine object detection with depth measurement, the algorithm has four nodes, each of which has a specific task.

The ZED-NODE is in charged of the depth data acquisition, provided by StereoLabs[2]. This is a node with many topics, but we use just two, i.e. an image publisher that contains a BGR image of 1280x720 pixels, and a depth image publisher, with the information of 921600 depth measurements, one per pixel. This matrix is a numeric array of float data in meters. Referring to Fig. 4, the ZED NODE, the ZED-GPU detection Node and the Depth Measurement Node were programmed in C++ while the OpenCV Node uses Phyton 3.6.

3 IMPLEMENTATION

In this section we show the process to embed the application into the NVIDIA Jetson Tx1. The process consists of three steps. The initial step is to generate a static library compatible with the GPU. The second step is to generate the ROS architecture shown in Fig. 4; this architecture consists of four nodes, of which 3 are used in this step; these are the depth acquisition "ZED Node" , the detection "Detection Node", and the image processing "OpenCV Node". The third step includes the fourth node, i.e. the "Depth Measurement NODE". Here we combine detection with depth measurement.

[1] https://pjreddie.com/darknet/yolov2/

[2] https://www.stereolabs.com/

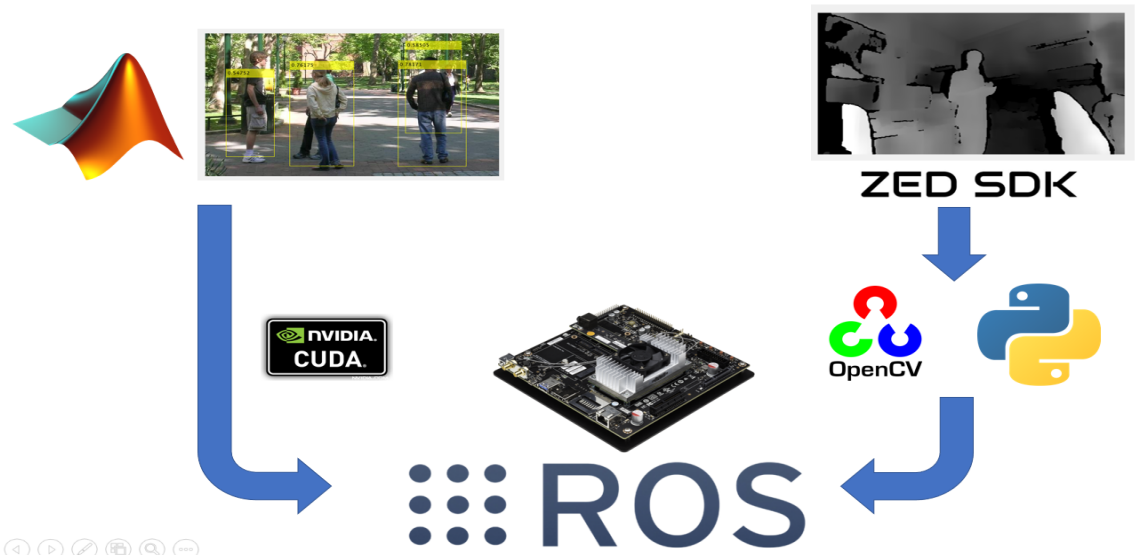

Figure 2: Object detection strategy combined with depth measurement.

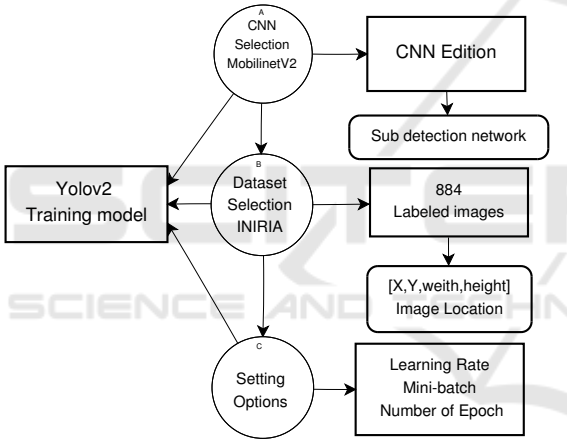

Figure 3: Flowchart to train YoloV2 detection network to detect the class person.

3.1 Static Library

We use the GPU coder in Matlab to generate a static library in the Jetson TX1. This library contains the object detector shown already in section 2. We built the static library based on a detection function, whose input is an image and the output are the bounding boxes, as presented in Fig. 5.

In addition, we use the cuBLAS[3] library which is an implementation of the Basic Linear Algebra Subprograms (BLAS). This library lets us access to the computational resources of the NVIDIA Jetson TX1 supercomputer. We use as well the cuDNN[4] library (Chetlur et al., 2014), which is a GPU-accelerated library for Deep Neural Networks (DNN). This library provides highly tuned implementations of common layer operations such as forward and backward convolution, pooling, normalization, and activation layers using the high performance of the Jetson TX1 module. Both, cuBLAS and cuDNN libraries are provided by NVIDIA with the purpose of optimizing the detection process and allowing to take advantage of the GPU. The detection library was configured to process an input image (1280x720 RGB column major image), as described in section 2.2. The ZED node publishes a 1280x720 BGR image. The OpenCV Node, converts the image into the format required by this library. After the detection process, the library returns the bounding boxes (bboxes) according to the detection process. These bboxes contain the information of the detected objects, which is presented in 4 variables, namely [X, Y, Width, Height].

The detection library was configured to process an input image (1280x720 RGB column major image), as described in section 2.2. The ZED node publishes a 1280x720 BGR image. The OpenCV Node, converts the image into the format required by this library. Then, it returns the bounding boxes (bboxes) according to the detection process. These bboxes contain the information of the detected objects, which is presented in 4 variables, namely

3.2 Image Processing

In this work we process the input image twice. First, in OpenCV, as shown in Fig. 4. The ZED-GPU DETECTION NODE receives a RGB image but the ZED NODE, publishes BGR images. For this reason we

[3]https://developer.nvidia.com/cublas Cublas
[4]https://developer.nvidia.com/cuDNN cuDNN

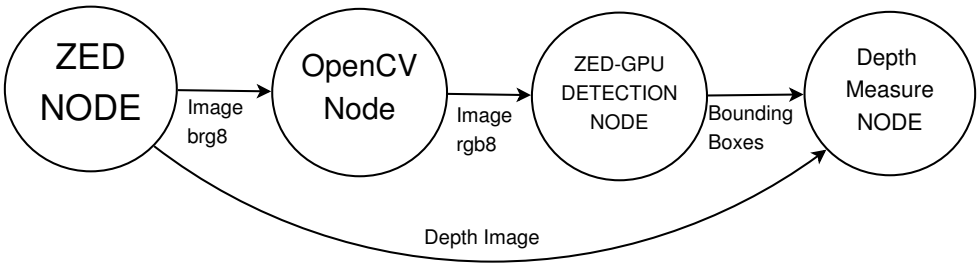

Figure 4: ROS diagram for depth measure.

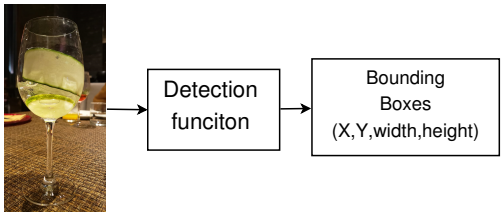

Figure 5: Proposed function to generate the static library.

propose another node to process the image to convert it into a RGB image. This reformatting process was done to avoid the deep neural network to be corrupted due to the color changing, which would result in a failure of the object detection. The second image processing was done to reformat this new RGB image (row major format) into an RGB (column major) image. This process is done because the static library needs an image formatted as 720x1280, while the ZED-NODE publishes it is 1280x720. Sending the wrong dimensions will not allow the algorithm to work properly.

3.2.1 Combining Object Detection with Depth Measurement

After processing the images, we carry out the object detection using the detection network presented in section 2, with the function structure illustrated in Fig. 5, and the static library proposed in section 3.1 using the NVIDIA Jetson TX1. This process is done on the ZED DETECTION NODE. The outputs of this node are the detected bounding boxes as Regions Of Interest (ROI) messages which consist of four data (X, Y, width, height). For example, in Fig. 6, the ROI message is printed. A,B,C and D are plots of the ROI messages given by the detection node. The combination method uses the ROI messages to set them as the dimensions of a new small image. Let us define the width and height of the new image, respectively as Im_w and Im_h, then these dimensions can be obtained

$$Im_w = X + W \tag{1}$$

$$Im_h = Y + H \tag{2}$$

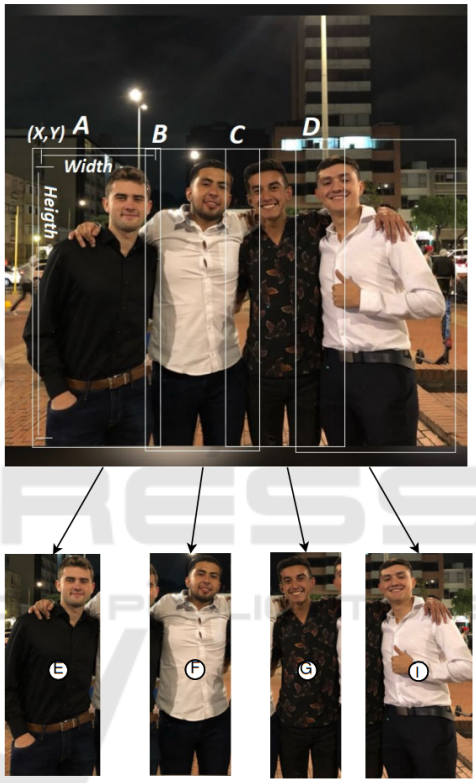

Figure 6: Image segmentation for depth measurement.

where X, Y, refer respectively to the (X,Y) coordinate in the original image, as it can be seen in Fig.6, and W, H are the width and height of the same image. Then, we measure the depth at the center of these new images, i.e. we obtain the depth of the points E,F,G and I, shown in Fig.6. This measurement corresponds to the distance between the detected person and the camera.

Once we segment the image with the ROI messages generated from the "DETECTION NODE", we receive and unpack the depth measurement subscribing the Depth measurement NODE to a depth topic of the ZED NODE. The distance is obtained in meters. At this point in the Depth measurement NODE there is a vector from the image depth topic of the distance of every single pixel and the ROI messages

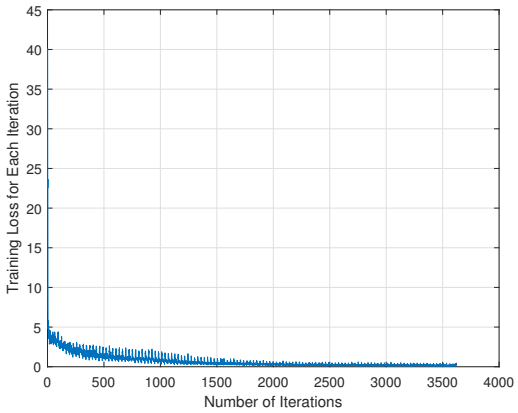

Figure 7: Training loss graph.

for the image segmentation. Finally, we use the linear indexing L_i to find the pixel desired position on the image representation, which means that we can refer to the elements of a matrix with a single subscript. In general, L_i is defined as

$$L_i = W * (C-1) + R, \tag{3}$$

where C and R are the desired column and row of the matrix that represents the image.

The important depth data for us is located in the middle of the new image result of the segmentation process described before. For this reason, we select the center pixel of the image through using (3), considering the pixel in the center of the image which is located at the point$(\frac{W}{2}, \frac{H}{2})$. Then, the linear index of the center pixel can be determined as

$$L_i = W * (C - \frac{H}{2}) + R - \frac{W}{2}. \tag{4}$$

4 RESULTS AND ANALYSIS

The training process with the INRIA and the PennFudanPed dataset lasted 90 minutes, using the host computer Intel corei5-7200 with 2 GB NVIDIA Geforce MX940. In Fig. 7 the training loss in the process is shown. The process consisted on 125 epochs, performing 3625 iterations, with a 30 Mini-batch size.

We carried out a validation test of the detection network, using the INRIA test set which contains 288 test images. In Table 1 we show the amount of data used for each training step. The tests consisted of the detection process of an image test set. The results of the detector, the scores, and the bounding boxes per image are compared with the ground truth proposed for the test set. The ground truth is a table with information about the location of the image on the computer where the test takes place and the bboxes of each

Table 1: Number of images per class.

Class	Train	Val	Trainval	Test
Person	614	170	784	288

image. According to the amount of data used to train the detector, we achieved an average precision of 72% in the host computer, and 84% in the NVIDIA Jetson Tx1.

The results of accuracy in detecting people, using the host computer and using the NVIDIA Jetson Tx1, are respectively presented in Figs.8 and 9. Fig. 9 shows the action of the cuBlas and cuDNN libraries which increases the detection accuracy in 12% on the Jetson with respect to the host computer.

During the experimentation process, we used a 60% threshold like the confidence of the network detection. It is worth mentioning that the stability of the camera is a crucial factor in the detection process, because with small disturbances the detection results in the sequence of images are noisy, so it is not useful. This issue is going to be solved in the next step of the project using a stabilizer system witch main purpose is to reduce the vibrations caused by robot displacement.

The robot platform in which this method will be implemented is the CERES agrobot shown in the Fig.1.The maximum speed reached by the system is 22.2 m/s, and its average speed during the experimental phase is 1.4 m/s. On the other hand, the detection average time is 0.21 s per image, and in order to make maneuvers or stop the platform, the stabilization time of the control system is around 2s, for this reason, the use of parallel computers is a solution to release the robot control system processor. This result implies that a robot as CERES can react to avoid collisions (3.5m is a safe distance for people from the robot, whilst only the system vision works, but the robot has other sensors that make safe its operation). For the purpose of our agricultural robot, decision making is possible to avoid damages to the robot and indeed to people or other objects that could appear suddenly as obstacles. However, it is imperative to find a solution to the stability in the detection process.

Regarding depth measurements, we compared practical real measurements with the results obtained from the data acquired. Fig. 10 illustrates the test carried out. It consisted on comparing the distance from the camera to three different objects placed at fixed known distances, at points A,B,C. We carried out 5 trials, with 3 different objects placed each time at a different point. For each trial we obtained the depth measurement using the Depth Measurement NODE of the algorithm. We compared the real measurement

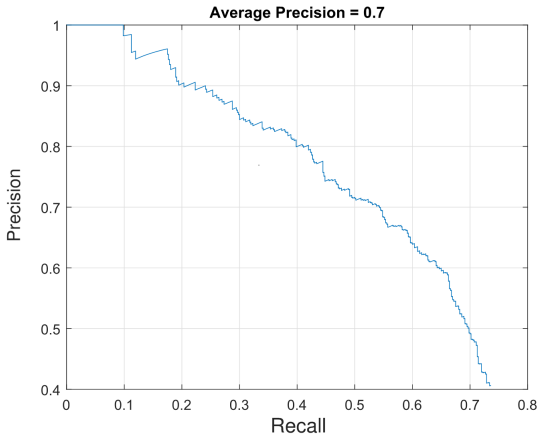

Figure 8: Average Precision of the detector using the Host Computer.

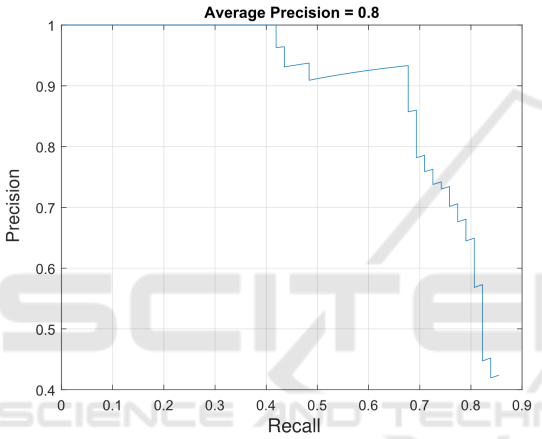

Figure 9: Average Precision of the detector using the NVIDIA Jetson module.

with the experimental measurement to evaluate the method. In this way, we obtain an average accuracy of 91%. In addition, the working range of the ZED camera is between $0.5m$ to $20m$, and the test has evaluated the accuracy of the measurements in this interval.

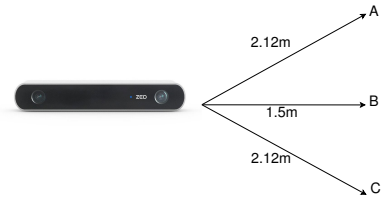

Figure 10: Depth measure test method.

5 CONCLUDING REMARKS

In this paper, we have proposed a to combine object detection with depth measurements for object tracking and decision making, for the agricultural robot CERES. In this case, we have used deep learning techniques with this purpose. Using a host computer, we achieve a detection network with an average accuracy of up to 72% in detecting the class "Person", while using a supercomputer Jetson, the accuracy increases up to 84%. The detection time is 0.21 s. These results are useful for our study because we can detect obstacles to prevent collisions and consequent damages to the robot and people. Furthermore, in our case, the vegetable farming process has beds (where the plants are sown) with 1m of wide x 50m of long, and furrows of 0.5m (where the robot tires can roll), thus we can accept that the decision making process can be carried out respecting to the mechanical system response (remember that the response time of the robot is about 2s), given that the robot's average speed is at least $1.4m/s$.

In future studies, we plan to train the detection algorithm with agricultural classes, such as undergrowth, flowers, plants, etc. not only to detect obstacles but also to help the robot to carry out the specific tasks efficiently.

ACKNOWLEDGEMENTS

This work is supported by the project INV_ING_3185 Sistema de toma de decisiones para la aplicacin de medidas correctivas que ayuden a mantener la salud de un cultivo de hortalizas utilizando un robot (CERES) dedicado a labores de agricultura financed by the Universidad Militar Nueva Granada in Bogot-Colombia.

REFERENCES

Ban, Y. and Lee, S. (2020). Protuberance of depth : Detecting interest points from a depth image. *Computer Vision and Image Understanding*, page 102927.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Breton, S., Quantin-Nataf, C., Bodin, T., Loizeau, D., Volat, M., and Lozach, L. (2019). Semi-automated crater depth measurements. *MethodsX*, 6:2293 – 2304.

Chen, Y., Zhao, D., Lv, L., and Zhang, Q. (2018). Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432:559 – 571.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., and Shelhamer, E. (2014). cudnn: Efficient primitives for deep learning. *CoRR*, abs/1410.0759.

Chlingaryan, A., Sukkarieh, S., and Whelan, B. (2018). Machine learning approaches for crop yield prediction

and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture*, 151:61 – 69.

Chua, S. N. D., Lim, S. F., Lai, S. N., and Chang, T. K. (2019). Development of a child detection system with artificial intelligence using object detection method. *Journal of Electrical Engineering & Technology*, 14(6):2523–2529.

Ciaparrone, G., Snchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., and Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61 – 88.

Ding, Y. and Xiao, J. (2012). Contextual boost for pedestrian detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2895–2902.

Espejo-Garcia, B., Martinez-Guanter, J., Prez-Ruiz, M., Lopez-Pellicer, F. J., and Zarazaga-Soria, F. J. (2018). Machine learning for automatic rule classification of agricultural regulations: A case study in spain. *Computers and Electronics in Agriculture*, 150:343 – 352.

Fujiyoshi, H., Hirakawa, T., and Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. *IATSS Research*, 43(4):244 – 252.

Girshick, R. (2015). Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Gong, Z., Lin, H., Zhang, D., Luo, Z., Zelek, J., Chen, Y., Nurunnabi, A., Wang, C., and Li, J. (2020). A frustum-based probabilistic framework for 3d object detection by fusion of lidar and camera data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:90 – 100.

Hendry and Chen, R.-C. (2019). Automatic license plate recognition via sliding-window darknet-yolo deep learning. *Image and Vision Computing*, 87:47 – 56.

Hu, J., Huang, J., Gao, Z., and Gu, H. (2018). Position tracking control of a helicopter in ground effect using nonlinear disturbance observer-based incremental backstepping approach. *Aerospace Science and Technology*, 81:167 – 178.

Kopp, M., Tuo, Y., and Disse, M. (2019). Fully automated snow depth measurements from time-lapse images applying a convolutional neural network. *Science of The Total Environment*, 697:134213.

Li, X., Zeng, Z., Shen, J., Zhang, C., and Zhao, Y. (2018). Rectification of depth measurement using pulsed thermography with logarithmic peak second derivative method. *Infrared Physics & Technology*, 89:1 – 7.

Li, Z., Dong, M., Wen, S., Hu, X., Zhou, P., and Zeng, Z. (2019). Clu-cnns: Object detection for medical images. *Neurocomputing*, 350:53 – 59.

Partel, V., Kakarla, S. C., and Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture*, 157:339 – 350.

Patrcio, D. I. and Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*, 153:69 – 81.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Reiss, D., Hoekzema, N., and Stenzel, O. (2014). Dust deflation by dust devils on mars derived from optical depth measurements using the shadow method in hirise images. *Planetary and Space Science*, 93-94:54 – 64.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

Sadgrove, E. J., Falzon, G., Miron, D., and Lamb, D. W. (2018). Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (mec-elm). *Computers in Industry*, 98:183 – 191.

Shin, J.-Y., Kim, K. R., and Ha, J.-C. (2020). Seasonal forecasting of daily mean air temperatures using a coupled global climate model and machine learning algorithm for field-scale agricultural management. *Agricultural and Forest Meteorology*, 281:107858.

Shinde, S., Kothari, A., and Gupta, V. (2018). Yolo based human action recognition and localization. *Procedia Computer Science*, 133:831 – 838. International Conference on Robotics and Smart Manufacturing (RoSMa2018).

Silva, J. V., de Castro, C. G. G., Passarelli, C., Espinoza, D. C., Cassiano, M. M., Raulin, J.-P., and Valio, A. (2020). Optical depth measurements at 45 and 90 ghz in casleo. *Journal of Atmospheric and Solar-Terrestrial Physics*, 199:105214.

Taiana, M., Nascimento, J. C., and Bernardino, A. (2013). An improved labelling for the inria person data set for pedestrian detection. In Sanches, J. M., Micó, L., and Cardoso, J. S., editors, *Pattern Recognition and Image Analysis*, pages 286–295, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., and Ma, X. (2020). 3d object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. *Sustainable Cities and Society*, 54:102002.

Wu, X., Sahoo, D., and Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*.

Zhao, Y., Mehnen, J., Sirikham, A., and Roy, R. (2017). A novel defect depth measurement method based on nonlinear system identification for pulsed thermographic inspection. *Mechanical Systems and Signal Processing*, 85:382 – 395.

Zhou, T., Ruan, S., and Canu, S. (2019). A review: Deep learning for medical image segmentation using multimodality fusion. *Array*, 3-4:100004.