Simulation of near Infrared Sensor in Unity for Plant-weed Segmentation Classification

Carlos Carbone[a], Ciro Potena[b] and Daniele Nardi
Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Rome, Italy

Keywords: Unity Engine, Near Infrared, Convolutional Neural Network, Plant-weed Segmentation Classification.

Abstract: Weed spotting through image classification is one of the methods applied in precision agriculture to increase efficiency in crop damage reduction. These classifications are nowadays typically based on deep machine learning with convolutional neural networks (CNN), where a main difficulty is gathering large amounts of labeled data required for the training of these networks. Thus, synthetic dataset sources have been developed including simulations based on graphic engines; however, some data inputs that can improve the performance of CNNs like the near infrared (NIR) have not been considered in these simulations. This paper presents a simulation in the Unity game engine that builds fields of sugar beets with weeds. Images are generated to create datasets that are ready to train CNNs for semantic segmentation. The dataset is tested by comparing classification results from the bonnet CNN network trained with synthetic images and trained with real images, both with RGB and RGBN (RGB+near infrared) as inputs. The preliminary results suggest that the addition of the NIR channel to the simulation for plant-weed segmentation can be effectively exploited. These show a difference of 5.75% for the global mean IoU over 820 classified images by including the NIR data in the unity generated dataset.

1 INTRODUCTION

Precision agriculture is the use of techniques that provide key information from plant crops to improve decision making about the use of the resources available in crop management (ISPA, 2020). In this area there has been an increase in research development in the recent years (Stafford, 2000; Khanal et al., 2017; Patrício and Rieder, 2018; Duhan et al., 2017; Carbone et al., 2018).

One of the main challenges in precision agriculture is to obtain the location of weeds that need to be removed to diminish possible quality loses in the main plants being grown (Lottes et al., 2018). The first step to obtain the location of such weeds is to have a reliable system that can identify them (Lottes et al., 2018). The most common system to accomplish this type of task is image-based classification through Deep Neural Networks. In particular, Convolutional Neural Networks (CNNs) are one of the methods that currently provide the most promising results for this purpose (Krizhevsky et al., 2017; Potena et al., 2017; Carvajal et al., 2017).

As any machine learning based method, CNNs require accurately labeled data to develop that can reliably identify plants and weeds in an image (Bah et al., 2018). The data required are usually images taken in the field in controlled environments. Acquisition of these data requires a lot of time and effort to then such data need to be manually labeled by hand (Bah et al., 2018). To ease such difficult task, simulation environments based on graphic engines have been developed to generate synthetic datasets for CNN training (Cicco et al., 2016). However, these simulators, to the best of our knowledge, have not included the simulation of the near-infrared (NIR) sensor, which have been proven quite effective on crop images (Milioto et al., 2017).

The objective of our research is to create a simulation environment that includes the near-infrared channel to generate a dataset for CNN training. This is accomplished through a sugar beet field simulated along with weeds in the Unity game engine (Xie, 2012), hereinafter referred to as "Unity", using textures gathered with a real world sensor. This simulator then generates images to train a selected CNN and compare its performance against the same net-

[a] https://orcid.org/0000-0001-5615-0225
[b] https://orcid.org/0000-0003-2395-2170

Carbone, C., Potena, C. and Nardi, D.
Simulation of near Infrared Sensor in Unity for Plant-weed Segmentation Classification.
DOI: 10.5220/0009827900810090
In *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2020)*, pages 81-90
ISBN: 978-989-758-444-2

work trained with a dataset of real images. Our experiments show that the performance result is improved by including the NIR data. Additionally, the simulator could be adjusted in future investigations for other types of plants. This would require additional textures and adjustments to the 3D meshes. The simulator could also be modified to include more points of views representing sensors mounted in different type of robots like ground robots or unmaned aerial vehicles (UAVs).

The rest of this paper is structured as follows: Section II presents the state of the art for plant-weed classification using CNNs and for simulation environments developed to generate CNNs training datasets. Section III presents the developed simulation environment in Unity along with texture gathering with a real sensor. Section IV explains the tests specifications. Section IV presents the selected convolutional neural network. Section V presents the results, the main key performance indicators and the discussion about the values obtained. Finally, section VI presents the conclusions and future works.

2 RELATED WORK

In this section we present the background state of the art by addressing first research developments for Plant-weed classification using convolutional neural networks and then for Simulation environments that aim to support data generation for deep learning researches.

2.1 Plant-weed Classification using Convolutional Neural Networks

The usage of UAVs in precision agriculture is one of the current commercial trends for these robots as well as for research approaches (Kolodny, 2017; Carbone et al., 2018). In (Sa et al., 2018a), an approach to achieve reliable plant-weed classification with UAVs hardware constraints was developed. SegNet was the network used with a Jetson TX2 integrated on the UAV. Efforts to include multispectral sensor data in convolutional neural network (CNN) training were further explored in (Sa et al., 2018b). Another approach that reduces the requirements of the network is presented in (Lottes and Stachniss, 2017). Here a semi-supervised approach is presented exploiting knowledge about the common arrangements of crops.

In (Lottes et al., 2018) a CNN was developed with an encoder-decoder structure that includes spatial information within sequences of images. This approach achieved a generalized improvement in results as it performs well on new unseen fields without the need to retrain the model. Considering spatial information is indeed a desirable feature in agricultural inspections, as they are often performed by robots that need to gather a sequence of images to cover the inspection of crops.

In (Fawakherji et al., 2019) an approach of using two networks in sequence is developed to do semantic segmentation classification. The first network being based on encode-decoder architecture to classify connected patches of plant instances from the soil, then the second network does the classification labeling. The results obtained were accurate enough in cases were the images had challenging features. Furthermore, (Li et al., 2019) make an approach in cases where the weeds are dense and overlap the plants, using ResNet-10 as base with the Adaptive Affinity Fields method.

2.2 Simulation Environments

This research builds upon the work "Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection" (Cicco et al., 2016), which proposed the use of images from a simulation environment to enhance the effectiveness of CNN for crop-weed classification using Unreal Engine 4 (Sanders, 2016), hereinafter referred to as "Unreal". The new feature of our research, which switches from Unreal Engine to Unity, is the inclusion of the simulation NIR data. Unity was chosen due to its better compatibility and accessibility on the Linux operating system. This will ease future research that requires robotics libraries that are mostly supported in Linux like the robotic operating system (ROS). To the best of our knowledge, the inclusion of the NIR data for plants in simulation environments based in graphics engines has not been done before. In particular, the research done in (Cicco et al., 2016) includes only the red-green-blue (RGB) data for synthetic dataset generation.

Video games have been used directly as source of training datasets for machine learning research. In (Richter et al., 2016), pixel output from the game as well as commands being issued were used to generate large datasets for machine learning research purposes. In (Shafaei et al., 2016) output images of cityscapes were generated and tested by comparing classification results of a CNN trained with the generated images and real images.

Further studies have included the customization of the environment simulated by using video game engines instead of video games with constraints defined by the game design. In (Kim et al., 2019), the Unreal

was used to simulate specific orientations of cars to train a model that works with video data from the real world; they highlight that the main problem in this domain is the lack of labeled imagery which can be generated automatically within Unreal. A similar approach was pursued with Unity in (Kaur et al., 2018) where the data was used to train a CNN as part of a system that provides drive assistance. Moreover, in (Yang et al., 2016), also develops a study in the car domain in Unity but with the focus of using the generated data to test sensor reaction instead of CNN training.

In (Acker et al., 2017) a cellular automation is implemented with Unreal to generate random pedestrian and vehicle movement to generate imagery that interfaces with a neural network through UnrealCV. The main idea was to generate an environment that had its movement established of an abstract level. Similarly, in (Hattori et al., 2015) a simulator for pedestrian behavior is developed. Similarly, in (Zhao et al., 2019) presents a Unity simulation for the training of a CNN for occupancy detection in a room which reduced the average localization error by 36.54% and 11.46% for private and public scenarios respectively using a real testbed.

In (Juliani et al., 2018) a platform for deep reinforcement learning is developed in Unity with the purpose of making available an open source simulation of learning agents in an environment with realistic visuals.

In (Akiyama et al., 2018) presents a method without a graphic engine that generates chart images to support studies in information visualization. A web tool is developed to synthetically and randomly generate the charts based on probability distributions functions. These charts are then rendered in PNG format.

The main contribution of this paper is the inclusion of the near infrared (NIR) data in a simulation environment developed in Unity that can build procedural fields of sugar beets and weeds including RGB and NIR data to generate images to be included as part of a CNN training dataset.

3 INTEGRATED SIMULATION ENVIRONMENT

In this research, Unity is the graphic engine used to develop the simulation environment to generate the synthetic dataset. Unity is designed to develop video games by providing scripting functionality along with realistic illumination. In Unity, we developed a realistic procedural crop field that randomizes parameters of its generated plants and weeds.

3.1 Sensor and Textures

Texture images and 3D meshes are required to generate the plants and weeds in Unity. Cropped images of real plants are used as textures which are cropped to fit into the 3D meshes through UV mapping which is the system that maps the pixels from the images to the 3D mesh. These images were gathered using the JAI AD-130 GE camera which provides input from a visible color channel from 400-700nm and a near infrared (NIR) channel from 750-900+nm simultaneously.

The plant images were taken in a small sugar beet field in the city of Ancona in Italy with the support of the personnel of the Agenzia Servizi al settore Agroalimentare delle Marche (ASSAM). In total, 8 textures were gathered, a cropped example image is shown in Figure 1. The weed textures were extracted from the 2016 Sugar Beets Dataset Recorded at Campus Klein Altendorf in Bonn, Germany, hereinafter referred to as "Bonn Dataset". A total of 50 weed textures were created by cropping the RGB and NIR images using the masks in the labeled images, an example is shown in Figure 2.

(a) RGB. (b) NIR.

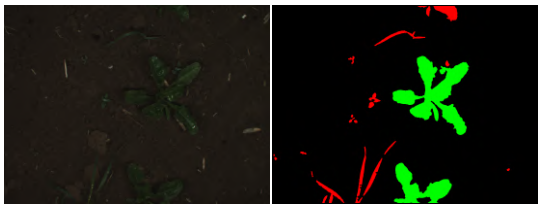
Figure 1: Sugar beet leaf textures, RGB on the left, NIR on the right.

3.2 Unity Engine

Once the texture images are obtained they must be placed in what is called materials in Unity with appropriate shader coding. The shader is the part of the simulator that takes the texture inputs and places them on top of the 3D meshes to render the appropriate pixels in the screen. Then the material is an instance of the shader code where textures images are specified.

Two main shaders are used in this simulation: one with transparency and double side rendering (used for plants and weeds), and one that shows raw colors to the screen excluding transparent pixels (used to create the masks for the labeled images).

For the plants, the textures are applied in a rectangular mesh with a skeleton that bends the mesh resembling a bent leaf. With the previously mentioned



(a) RGB. (b) Labeled.

(c) Cropped.

Figure 2: Weeds in sugar beet field, original RGB on the left, labeled image on the right, Green labels are the plants and red labels are the weeds, and texture showing only the weeds cropped at the bottom.

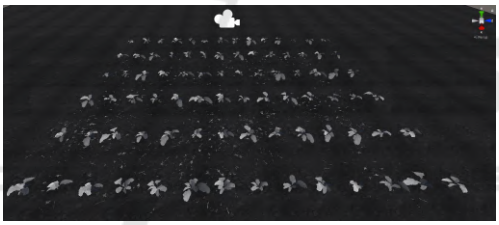
shaders only the leaf pixels are visible in the mesh, visually turning the bent rectangle to a leaf. With the same shaders, the weed textures are placed in a simple square due to the very small shapes of the weeds.

With the plant and weed meshes ready to be spawned, a script is created to procedurally generate the field with some randomized parameters to create a high variety of plants and weeds. The size and rotation of the sugar beet leaves have random variations to have similar shapes compared to the real images. The weeds are randomly placed as planes on the ground over the field with their rotation being randomized. A field is shown in Figure 3, and images taken from the camera in the generated level are shown in Figure 4, the RGB taken images are also shown in Figure 5 where the brightness has been manually adjusted for demonstration purposes for this paper. During the image generation the following steps are taken repeatedly until the desired amount of images is generated:

1. The camera is placed in a fixed position it takes a screenshot of the field setup with RGB data.
2. The field switches to the NIR data by changing the Unity materials and the camera takes another screenshot.
3. The field switches to the label mask data by changing the Unity materials and the camera takes another screenshot.
4. The intensity of the global illumination is slightly varied with random values, the field is replaced with a completely new one and the camera is moved to a new random position within the crop.
5. Above steps are then repeated to generate more images.

The close up images present a side by side comparison of the Unity synthetic images with the real images of both RGB and near infrared data. The Unity images show similar features, plant sizes and illumination that are essential to get good classification results for real images. This allowed to generate a synthetic dataset of plant-weed images including the RGB view, the NIR view and the labeled view that holds a similar aspect to the real dataset. The specific properties of the datasets built will be introduced in Section 4. Additionally, the dataset can be accessed in https://github.com/CSCarbone07/SPQR_AgriSim_Unity.

(a) Unity RGB field.



(b) Unity NIR field.

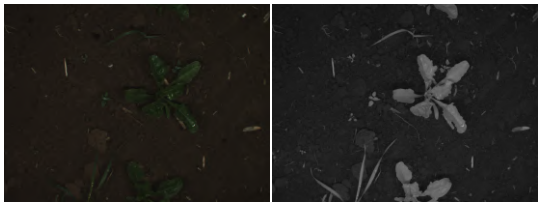
(c) Unity labeled masks field.

Figure 3: Sugar beet leaf field: RGB (above), NIR and labeled masks (below).

3.3 Convolutional Deep Learning Neural Network

The deep neural network used is: ”Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics” (Milioto and Stachniss, 2019), hereinafter referred to as ”Bonnet”. This network is used with its default configuration for plant-weed classification, and some modifications

(a) Unity RGB image. (b) Unity NIR image.



(c) Real RGB image. (d) Real NIR image.

Figure 4: Images taken from camera for synthetic dataset (above) compared to real images (below): RGB to the left, NIR to the right.

were done to the code to include a fourth channel to include the input NIR data. The research reported a mean intersection over union (IoU) of 80.1% for crop-weed classification and 98.5% for the mean accuracy. The code is also adapted to be mounted on robots with ROS if necessary since the network was released as an open source tool to ease its usage on robotics applications.

The network is based on an encoder-decoder architecture, and the plant weed configuration has a 9 layer configuration with 3 down-sample blocks of 8, 16 and 64 filters respectively and 3 blocks of up-sample of 32, 16 and 8 filters respectively. This configuration setting was developed to work with the real data set that is used in this research, Bonn Dataset (Chebrolu et al., 2017), which is the dataset used for testing in this research.

4 TESTS

This section explains the key features of the datasets used for the training of the deep learning neural network. A summary of these features is presented in Table 1.

The synthetic dataset was generated using the level built in Unity, creating 1034 images with a size of 1024x1024 pixels, which is one of the common options for Unity textures. The mentioned amount of images was generated to have an amount of images close to the half of the images available in the Bonn Dataset, which are used as the source of real images

(a) Unity RGB enlightened image.

(b) Real RGB enlightened image.

Figure 5: Enlightened RGB images from Figure 4.

for training and performance evaluation. This dataset has a total 1854 images, where 1034 are used as the real dataset for the training of the CNN by fitting them into 734 for the train images, 150 for the validation images and 150 for the test images as shown in Table 1. This leaves 820 real images which were used for the performance evaluation of each trained CNN using each dataset. The Unity synthetic dataset has the same distribution of images as the real dataset for the training of the CNN. Then, a mixed dataset is built where 300 image of the real dataset are added and distributed equally for the training, validation and test along with the synthetic dataset distribution for the CNN training as shown in the third column in Table 1.

Each dataset built has an RGB and an RGBN version and both share the same distribution of images. The labels are defined as ground, weed and plant in all the datasets. All the textures were gathered with the same sensor model used to take the real images in the Bonn Dataset. The pixels for the real dataset training

Table 1: Datasets features.

Feature for RGB and RGBN input	Real	Unity	Unity + Real
Dimensions (WxH pixels)	1296x966	1024x1024	512x512
Train (number of images)	734	734	734 Unity +100 real
Validation (number of images)	150	150	150 Unity +100 real
Test (number of images)	150	150	150 Unity +100 real

are the same as the sensor output, for the Unity dataset the output of 1024x1024 pixels is used, and for the mixed dataset the images were reduced to 512x512 pixels to have the same dimensions. Both the synthetic and the mixed dataset have a power of two size of images which is a common property for Unity images. The each training using each dataset had at least 200 epochs with the aim of running the training until the gain in accuracy was negligible.

5 RESULTS

The results for each training are presented using the IoU and the accuracy as key performance indicator (KPI). The former is calculated through Equation 1 for each label: N is number of pixels of a classified image, T_{pi} is the true positive where a pixel is classified correctly for a given label. The sum of these being equivalent to the intersection of the classified image with the ground truth image for the chosen label. Then, the T_{pi}, F_{pi} and F_{ni} are the True positive, False positive and False negative respectively for a label. Then the IoU mean is obtained by calculating an average of the IoU mean for each label on a given image, hence the total IoU mean is calculated by taking a mean over all the IoU means of the 820 classified images for performance evaluation.

$$IoU_{label} = \sum_{i=1}^{N} \frac{T_{pi}}{T_{pi} + F_{pi} + F_{ni}} \quad (1)$$

The second KPI used is the accuracy which is calculated using Equation 2. Again, this is calculated for each label and then a mean is calculated per image and a mean over the 820 images is calculated as well for this performance evaluation. The variables are the same ones in Equation 1 with the addition of the T_{ni} which is the true negative when a pixel does not contain a certain label in both the classified image and the ground truth. Each KPI result is then multiplied by 100 to have the results in percentages.

$$Accuracy_{label} = \sum_{i=1}^{N} \frac{T_{pi} + T_{ni}}{T_{pi} + T_{ni} + F_{pi} + F_{ni}} \quad (2)$$

Table 2 and 3 were obtained by using the selected KPIs. The former represents the overall result after calculating the IoU and accuracy means for the classification over 820 labeled images using RGB (red-green-blue) and RGBN (red-green-blue-near infrared) inputs from Bonn Dataset (Chebrolu et al., 2017). The results of table 2 are also presented in Figure 6 and 7 for better visualization. Table 3 presents the KPIs of interest for the image that got the highest difference between RGB and RGBN mean IoU using the synthetic (Unity) trained network. The columns represent each classification result for said image using each trained CNN with each dataset. This image is shown in Figure 8, which also presents the classification ground truth (plant green and weed red labeled image) and the result classification (plant green and weed blue labeled image) using the trained network for RGB and RGBN inputs using the 3 training datasets: real, Unity and Unity+real.

5.1 Discussion

As it can be seen in Table 2 and 3, an increase in mean IoU and mean accuracy is achieved by adding the input of the NIR channel for the deep learning network training. As an overall, the mean IoU and mean Accuracy (lines 8, 9, 16 and 17 of the table) has a difference of 5.75% and 0.64% respectively for the trained neural network using the Unity training dataset shown in the third column of Table 2. The same trend is found in the classification results using the other trained networks using the real and Unity+real dataset, second and fourth column respectively. This means that the NIR data can increase considerably the classification results, specially for the synthetic dataset as this presented the biggest improvement compared to the other dataset trained networks.

Table 2 also includes the results for each class: Plant, Weed, Ground. In these lines (2-13), it can be seen that the best performance is for the ground followed by the plants and then the weeds, which follows an order of pixels. This is because the ground data is considerably different from the other two classes. Additionally, there was more training and classification pixel data available for the ground and less for plants and weeds. Therefore, the classification results are in favor of soil ground classification compared to plant and weed classification. Additionally, it is worth mentioning that the available labeled images from the

Table 2: Classification mean results (in percentage) for classification over 820 images used for evaluation performance.

Method	Real	Unity	Unity + Real
Plant IoU RGB	83.35	60.83	73.51
Plant IoU RGBN	83.07	71.20	75.61
Weed IoU RGB	50.49	22.54	31.57
Weed IoU RGBN	54.09	29.57	36.82
Ground IoU RGB	98.56	98.14	98.41
Ground IoU RGBN	98.53	97.97	98.73
mean IoU RGB	77.47	60.50	67.83
mean IoU RGBN	78.56	66.25	70.38
Plant Accuracy RGB	99.04	97.23	98.63
Plant Accuracy RGBN	99.18	98.33	98.70
Weed Accuracy RGB	98.93	97.75	98.59
Weed Accuracy RGBN	99.03	98.18	98.70
Ground Accuracy RGB	99.99	99.97	99.99
Ground Accuracy RGBN	99.99	99.97	99.99
mean Accuracy RGB	99.32	98.18	99.07
mean Accuracy RGBN	99.40	98.82	99.16

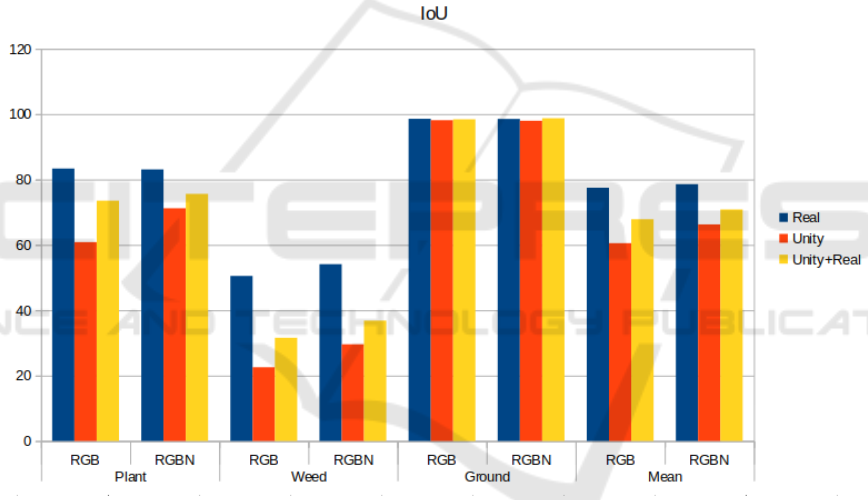


Figure 6: IoU results from table 2 for each class.

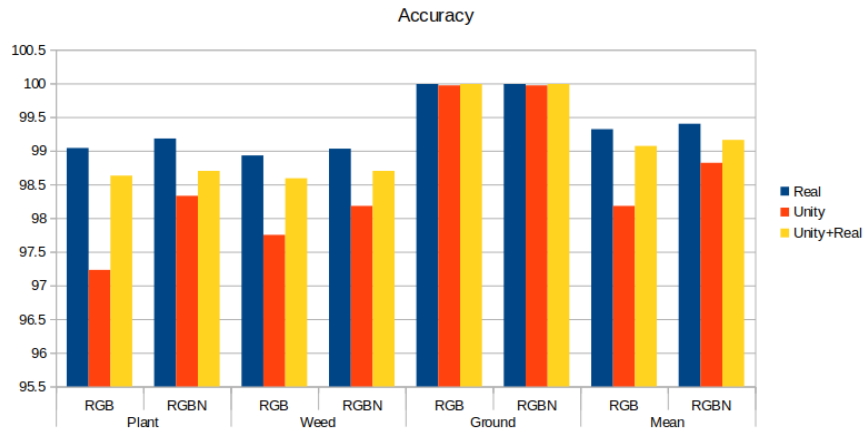
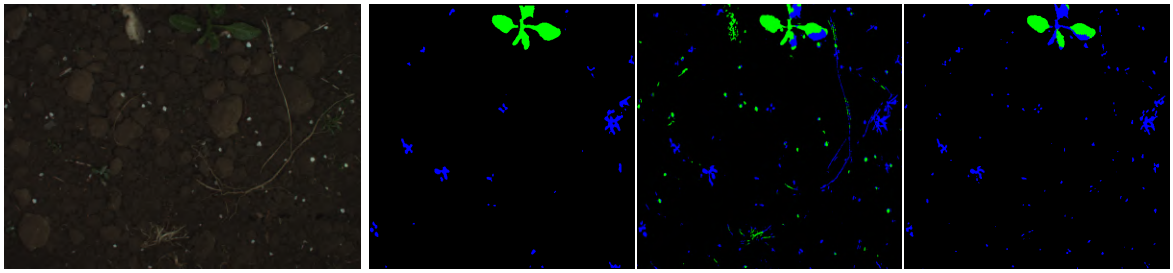


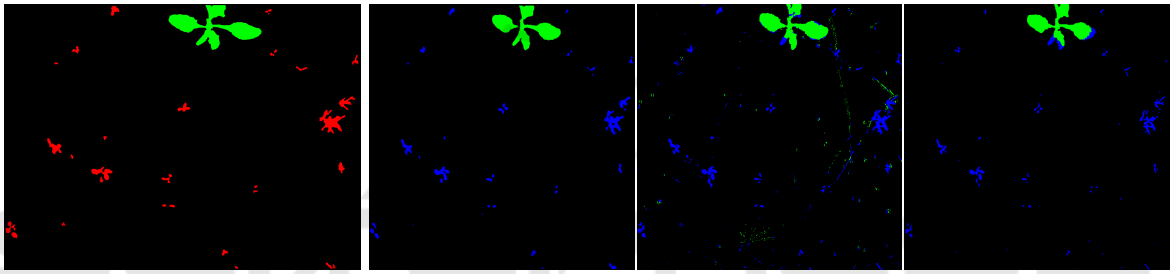
Figure 7: Accuracy results from table 2 for each class.

Table 3: Classification results (in percentage) for selected image with a high difference between RGBN and RGB mean IoU score using the Unity trained network.

Method	Real	Unity	Unity + Real
mean IoU RGB	81.56	57.68	59.45
mean IoU RGBN	83.38	74.03	71.93
mean Accuracy RGB	99.76	99.14	99.29
mean Accuracy RGBN	99.79	99.58	99.62



(a) Real RGB image. (b) Real (RGB). (c) Unity (RGB). (d) Unity+real (RGB).



(e) Real image ground truth. (f) Real (RGBN). (g) Unity (RGBN). (h) Unity+real (RGBN).

Figure 8: Classified image with high difference between RGBN and RGB mean IoU using the Unity trained network. First column present original image and ground truth labeled image. Second column presents the classification result using the real dataset trained network. The third column presents classification result for the Unity trained network. And the fourth column presents the classification result for the Unity+real trained network. Green labels plants, red labels weed in ground truth and blue labels weeds in network classified image.

Bonn Dataset are in a slightly younger state compared to the used for the synthetic generated images, which also creates spread results compared to previous researches (Cicco et al., 2016).

The mean IoU results obtained using the real images, shown in the second column of Table 2, are in line with previous reported results for plant-weed classification using Bonnet, these are 80.1% for the mean mIoU and 98.5% for the mean accuracy (Milioto and Stachniss, 2019). It is worth noting that the difference of mean IoU between the RGB and RGBN for the real dataset training is of 1.09% and 0.08% for the mean IoU and mean Accuracy respectively could be considered small. Multiple trends for these values have been found by previous researches when the NIR is added as an input for plant-weed classification. (Milioto et al., 2017) reported multiple results with differences of mean IoU ranging between 2.98% and 27.94% and (Deng et al., 2014) reports a negative difference of 26.27%, meaning that there are cases where the extra data from the NIR could bring confusion to the neural network.

Table 3 and Figure 8 show a case where the NIR makes the big improvement for plant-weed classification. Here the biggest change of interest is the mean IoU where a difference of 2.65% and 18.7% is achieved for the real dataset trained network and Unity trained network respectively.

As both Table 2 and 3 suggest, the improvement trend by including the NIR input data is more prominent in the synthetic dataset method, meaning that this contribution is remarkable for simulation environments developed to generate synthetic dataset.

In previous research "Automatic Model Based Dataset Generation for Fast and Accurate Crop and Weeds Detection" using Unreal as the simulation environment to generate the synthetic dataset results obtained are up to 60.2% and 99.5% for the mean IoU and mean accuracy respectively (Cicco et al., 2016). This result was obtained by using also a set of the im-

ages included in the Bonn Dataset (Chebrolu et al., 2017) using a modified version of the RGB SegNet as the convolutional neural network (CNN) with a synthetic generated dataset of RGB images resized to 480x360 pixels (Badrinarayanan et al., 2015).

6 CONCLUSIONS AND FUTURE WORK

The main contribution of this paper is the addition of the near infrared sensor in a simulation environment to generate datasets of plant-weed images that include red, green, blue and near infrared data. The results were tested against real data with a convolutional neural network (CNN) designed for plant-weed segmentation classification. The classification is evaluated using the mean intersection over union (IoU) and the accuracy: both values are increased by adding the near infrared data, with the most prominent improvement obtained by using the synthetic data.

The results obtained suggests this method to be bases for the simulation of other type of plants and weeds with the purpose of providing a reliable source of data to generate datasets for CNN training. Simulation parameters as size of plants, illumination intensity and color variations can be adjusted to fit a specific environmental condition, and thus achieve similar classification results. Additional type of plants and weeds can be added as well by adding new textures or new 3D meshes to depending on their complexity. If the new plant general shape is the same, it would require only additional textures and simple adjustments to the 3D meshes. The developed simulator generates sugar beets leaves using a simple rectangular bent mesh which then take the shape of the leaf through a transparency shader that only renders the pixels with leaf pixels. Then, carrots, for example, could be easily added since they have also a bent stem with leaves that follow to some degree the pattern of the bent stem. For some studies simple new textures could be enough and if more detail is required then extra bent meshes could be spawned on top of the main stem to have more precise leave shapes. Plants with considerable different shapes like sunflowers could also be added by adding appropriate meshes that follow the general shape of the plant and their required textures.

The simulator could also be modified to include more points of views representing sensors mounted in different type of robots like ground robots or UAVs. These robots could also be added in the simulator to perform inspection patterns and collect data for machine learning training and testing.

REFERENCES

Acker, O. V., Lachish, O., and Burnett, G. (2017). Cellular automata simulation on FPGA for training neural networks with virtual world imagery. volume abs/1711.07951.

Akiyama, R., Araujo, T., Chagas, P., Miranda, B., Santos, C., Morais, J., and Meiguins, B. (2018). Synthetic chart image generator: An application for generating chart image datasets. In *2018 22nd International Conference Information Visualisation (IV)*. IEEE.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561.

Bah, M., Hafiane, A., and Canals, R. (2018). Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sensing*, 10(11):1690.

Carbone, C., Garibaldi, O., and Kurt, Z. (2018). Swarm robotics as a solution to crops inspection for precision agriculture. *KnE Engineering*, 3(1):552.

Carvajal, J. A., Romero, D. G., and Sappa, A. D. (2017). Fine-tuning based deep convolutional networks for lepidopterous genus recognition. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 467–475. Springer International Publishing.

Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., and Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36(10):1045–1052.

Cicco, M. D., Potena, C., Grisetti, G., and Pretto, A. (2016). Automatic model based dataset generation for fast and accurate crop and weeds detection. volume abs/1612.03019.

Deng, W., Zhao, C., and Wang, X. (2014). Discrimination of crop and weeds on visible and visible/near-infrared spectrums using support vector machine, artificial neural network and decision tree. *Sensors & Transducers*, 26:26–34.

Duhan, J. S., Kumar, R., Kumar, N., Kaur, P., Nehra, K., and Duhan, S. (2017). Nanotechnology: The new perspective in precision agriculture. *Biotechnology Reports*, 15:11–23.

Fawakherji, M., Youssef, A., Bloisi, D., Pretto, A., and Nardi, D. (2019). Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE.

Hattori, H., Boddeti, V. N., Kitani, K., and Kanade, T. (2015). Learning scene-specific pedestrian detectors without real data. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3819–3827.

ISPA (2020). Home | international society of precision agriculture.

Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mat-

tar, M., and Lange, D. (2018). Unity: A general platform for intelligent agents. *CoRR*, abs/1809.02627.

Kaur, P., Sobti, R., and Khamparia, A. (2018). Simulation and deep cnn based architecture for validation of intelligent automotive functions. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2344–2348.

Khanal, S., Fulton, J., and Shearer, S. (2017). An overview of current and potential applications of thermal remote sensing in precision agriculture. *Computers and Electronics in Agriculture*, 139:22–32.

Kim, T. S., Peven, M., Qiu, W., Yuille, A., and Hager, G. D. (2019). Synthesizing attributes with unreal engine for fine-grained activity analysis. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 35–37.

Kolodny, L. (2017). Fixed-wing drones not quite taking off in commercial market, a new DroneDeploy study finds \textbar TechCrunch. https://techcrunch.com/2016/08/15/fixed-wing-drones-not-quite-taking-off-in-commercial-market-a-new-dronedeploy-study-finds/. Visited on 2017-08-09.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Li, N., Zhang, X., Zhang, C., Guo, H., Sun, Z., and Wu, X. (2019). Real-time crop recognition in transplanted fields with prominent weed growth: A visual-attention-based approach. *IEEE Access*, 7:185310–185321.

Lottes, P., Behley, J., Milioto, A., and Stachniss, C. (2018). Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters*, 3(4):2870–2877.

Lottes, P. and Stachniss, C. (2017). Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Milioto, A., Lottes, P., and Stachniss, C. (2017). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. *CoRR*, abs/1709.06764.

Milioto, A. and Stachniss, C. (2019). Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.

Patrício, 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.

Potena, C., Nardi, D., and Pretto, A. (2017). Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *Intelligent Autonomous Systems 14*, pages 105–121. Springer International Publishing.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 102–118, Cham. Springer International Publishing.

Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J., and Siegwart, R. (2018a). weedNet: Dense semantic weed classification using multispectral images and MAV for smart farming. *IEEE Robotics and Automation Letters*, 3(1):588–595.

Sa, I., Popović, M., Khanna, R., Chen, Z., Lottes, P., Liebisch, F., Nieto, J., Stachniss, C., Walter, A., and Siegwart, R. (2018b). WeedMap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming. *Remote Sensing*, 10(9):1423.

Sanders, A. (2016). *An Introduction to Unreal Engine 4*. CRC Press.

Shafaei, A., Little, J. J., and Schmidt, M. (2016). Play and learn: Using video games to train computer vision models.

Stafford, J. V. (2000). Implementing precision agriculture in the 21st century. *Journal of Agricultural Engineering Research*, 76(3):267–275.

Xie, J. (2012). Research on key technologies base unity3d game engine. In *2012 7th International Conference on Computer Science Education (ICCSE)*, pages 695–699.

Yang, C., Lee, T., Huang, C., and Hsu, K. (2016). Unity 3d production and environmental perception vehicle simulation platform. In *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pages 452–455.

Zhao, J., Frumkin, N., Ishwar, P., and Konrad, J. (2019). Cnn-based indoor occupant localization via active scene illumination. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2636–2640.