Polytechnic University of Madrid

HIGHER TECHNICAL SCHOOL OF ENGINEERING AND TELECOMMUNICATION SYSTEMS

CARROT DETECTION JETSON NANO

DTE - Department of Telecommunications and Electronics

Author:

Alberto Sánchez

May 2025

Contents

List of Figures

List of Tables

1 Introduction

The objective of this paper is to show the study done on Edge AI and to formalize a possible use case to be deployed in real-life systems.

To focus on developing the system, it is necessary to understand the background of this concept and similar projects that have been studied in this field of edge AI related to precision agriculture.

Edge AI seems to be the solution to cases where fast inference, scalability, saving costs and privacy is essential. Precision agriculture is known to be new feasible solutions in today's world.

Due to the specificity of the DTE department in precision agriculture, research has been focused on some solution that can help farmers identify crop problems and solve them as soon as possible.

The solution that will be given is to maintain the quality of the crop as best as possible, with an early diagnosis, so that more incomes and less food will be discarded.

The previous case and the one developed are designed to be deployed in the Jetson Nano, which is the edge AI component responsible of processing and making the real-time inference.

The previous project will explain the problem and the modeling of the AI. The use case, carrot detection (system deployed on the Jetson Nano) while the AI algorithm shows from the internal architecture, the training and validation steps and finally the section of how to execute the model inside the Jetson Nano.

2 Previous Projects

2.1 Agricultural Crop Image Classification

Firstly, a very easy task to know what kind of classification could be done in relation to precision agriculture was the classification of agricultural crops. [1] It consists of differentiating 30 different types of crops, such as rice, sugar, maize, lemon, banana, and coconut, among others. Only data sets were found; no paper.

2.2 Corn Leaf Infection Detection

Consists of an open source data set [2] in order to help develop the agricultural sector by using these images to create a system capable of detecting infected corn leaves by some pests, like for example the Fall ArmyWorm. Segmentation models would fit this use case.

2.3 AI and ML for ATV Farm Automation

In this article [3], some practical applications are shown, from improving sowing and planting operations to detecting weeds and controlling the de spray herbicide task. Other options are precise spraying and fertilization and forecasting production and indentifying the best time to harvest.

2.4 FlexiGrobots

Performing more research, the **FlexiGrobot** project was taken into account. FlexiGrobot is a European Union project of precision agriculture funded by the European Commission on the Horizon 2020 program. It is a consequence of a large study in which an AI algorithm was used to detect unhealthy grape brunches, more specifically Botrytis fungus, in different vineyards. They used to types of vehicles UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles). The process consists of four steps:

1. **Aerial Exploration**: UAVs perform aerial inspections and predict where there is a high probability of Botrytis in the vineyards. This prediction was performed using a random forest algorithm in which the inputs were tree top height (CHM), normalized difference of vegetation index (NDVI), and leaf area index.
2. **Ground Exploration**: UGVs travel through the areas predicted by the UAVs and take pictures of the corresponding vineyard. In this case, the images are sent to a data center for confirmation. They can also be detected in real time if computationally necessary.
3. **Treatment**: Robots travel through areas affected by this disease and apply phytosanitary products according to the map or if they are detected in real time.
4. **Assist**: A set of AGVs also assists operators in manually harvesting the crop, creating a cooperative work environment to carry the baskets.

In this case, what is of our interest is the IA model used to predict the Botrytis fungus, that is to say, the UGVs model. The UGVs takes the picture of the vineyard and firstly identifies the bunches of grapes with a bounding box and finally classifies if there's the fungus.
The IA model consists of a CNN (Convolutional Neuronal Network), which extracts the characteristics and draws the bounding boxes, and the final layer, the classifier, assigns the bunches of

grapes as healthy or unhealthy. The camera used in the Jetson Nano was a **Canon EOS 7D**, which is compatible with the module of the Developer Kit E used in this study. A visual of the classification of the grapes should be like this.

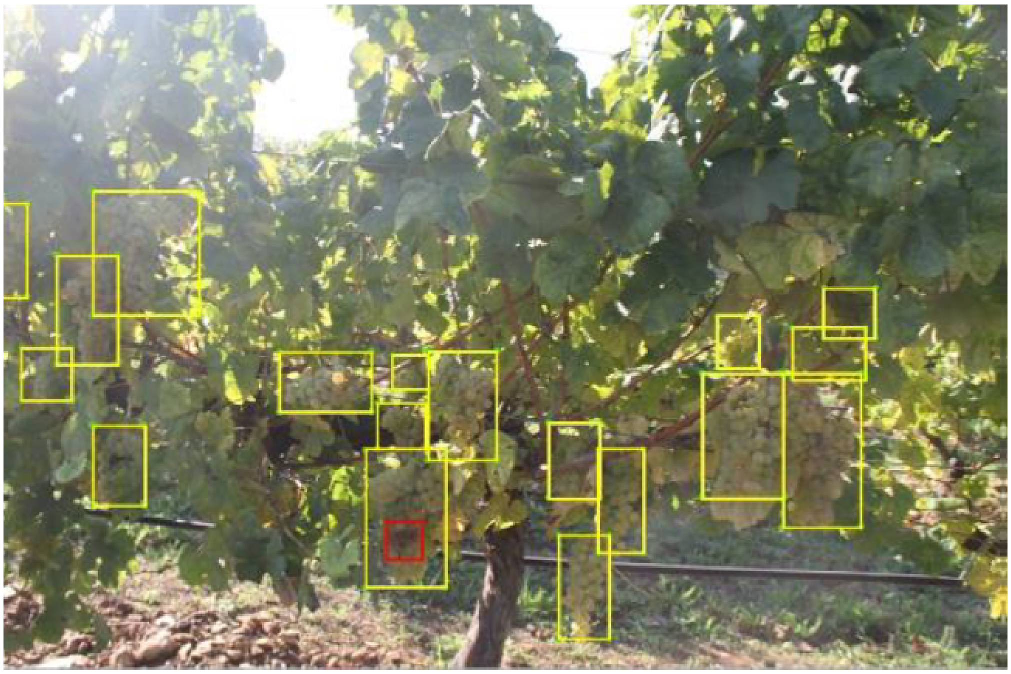

Figure 1: Labeled Grapes

All of the different models that were used consisted as said before of different types of CNN, which using filters and convolutions, the extraction of characteristics such as texture, color or edges among other things are useful to detect and then a MLP (Muti Layer Perceptron), more common known as basic Neuronal Network classifies using a softmax function between class 1 (RD) or 0 (RS), corresponding to unhealthy and healthy respectively.

The different models used were:

Table 1: Models and Input Config

Architecture	Version	Model	Input Image
FRCNN	-	ResNet50	640x640
	-	ResNet101	640x640
	-	ResNet152	640x640
EfficientNet	D0	EfficientNet	512x512
	D1	EfficientNet	640x640
	D2	EfficientNet	768x768
	D3	EfficientNet	768x768
CenterNet	-	ResNet50	512x512
	-	Hourglass 104	512x512
SSD	-	MobileNet V1	640x640

And the different metrics that evaluated the precision of the model to classify the different grapes correctly were mean IoU (Intersection Over Untion) was selected as the metric.

Table 2: Models and Input Config

Architecture	Backbone	Input Size	Steps	mAP @ 0.5IoU
FRCNN	ResNet50	640x640	43k	0.86
	ResNet101	640x640	35k	0.73
	ResNet152	640x640	20k	0.75
EfficientNet	EfficientNet	512x512	54k	**0.88**
	EfficientNet	640x640	27k	0.81
	EfficientNet	768x768	38k	0.83
	EfficientNet	768x768	25k	0.80
CenterNet	ResNet50	512x512	5k	0.75
	Hourglass 104	512x512	8k	0.78
SSD	MobileNet V1	640x640	7k	0.71

So, the idea was to recreate this project and watch that the different models used were feasible as the article shows, and, if they were, try to achieve better results in terms of inference or accuracy prediction. The data sets were published in the CSIC repository [4]. The fact was that only unhealthy grapes were in the data set and there was no label on which bunch of grapes were healthy or not. In the FlexiGrobots paper [5] it was said that the labeling process was carried out by experts, so it was difficult to attack this problem.

In fact this reason, other use of case was search to take another precision agriculture scenario.

3 Use Case

After searching for different possible use cases, the detection of carrots and brushes was found. It consists of a dataset that can be a simulation, as the Jetson Nano took the pictures, where at the very first part of the crop, a segmentation model using IA can predict the healthy carrot and the brush which is the unhelathy part.

The brush is the part of the crop that contains several fungi, and the consequence is that the healthy part absorbs less water, which is essential to grow in a correct way. Also, the main idea is that the decision part that is going to be deployed is of help to the herbicide system. The herbicide tank is integrated in the UGVs along with the Jetson Nano, which carries the processing part.

The main objective is to detect where the herbicide tank should spray the herbicide and thus reduce the unhealthy part as soon as possible without the absence of the farmer and optimizing the amount of herbicide used. With this solution, Jetson Nano takes the responsibility of the end device in Edge AI.

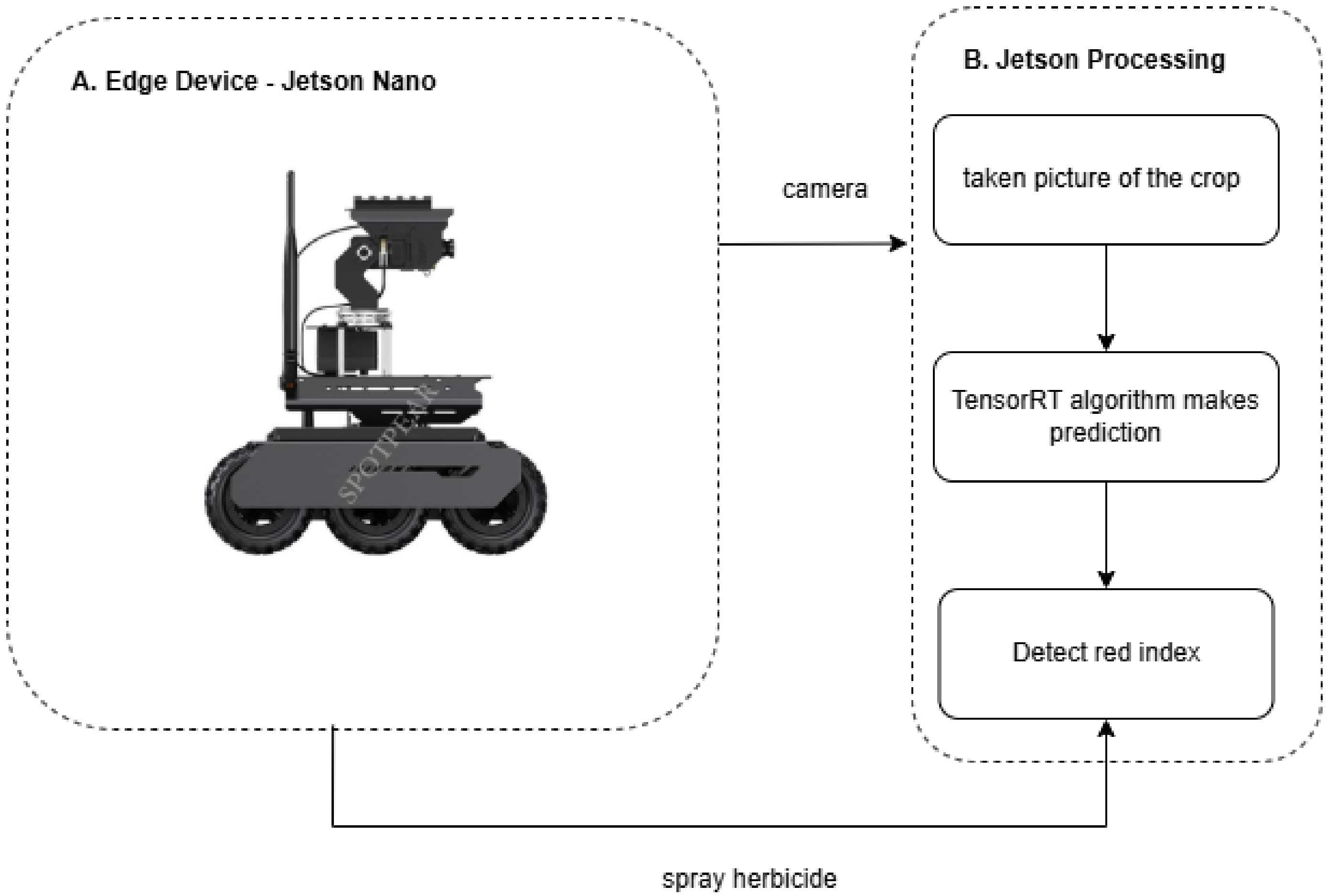

Figure 2: Diagram Block

This diagram block represents the general idea of the work system, however, later on the real architecture will be shown. In this paper, the herbicide tank is explained as a real life solution, nevertheless, the main objective is to build an AI capable of detecting carrots and weeds.

4 AI algorithm

4.1 Architecture

Knowing the use of the case and general operation, the AI algorithm takes place. Firstly, it is essential to identify the type of task that is going to be performed, that consists of a semantic segmentation. Semantic segmentation consists of a computer vision algorithm in which a neuronal network classifies each pixel into a different class. There are many different semantic segmentation architectures, and the most usuals are FCCN (Fully Convolutional Network), DeepLab, PSPNet

(Pyramid Scene Parsing Network) or BiSeNet among others. In the paper related to the study they used a DeepLab architecture, however, the **U-Net** model was chosen.
The reasons for using U-Net instead of DeepLab where:

1. **Real Time Performance**: Although DeepLab may achieve better results, the difference is not very significant. Instead, real-time inference is much faster using U-Net.

2. **Scalability**: Jetson Nano has limited resources, so it is important to deploy a model compatible and stable, for that reason U-Net is better.

3. **Compatibility**: DeepLab introduces advanced components such as atrous(dilated) convolutions and ASSP (Atrous Spatial Pyramid Pooling), which can highly cause compatibility issues when exporting models and running them in limited support hardwares, such as Jetson Nano. Also, U-Net algorithm may be capable of being trained inside Jetson Nano core.

4. **Libraries incompatibilities**: Due to the complex architecture some incomptabilities of library versions were given and to solve them it was very complex, therefore, it was decided to opt for a model that meets the objective and capabilities of the device well.

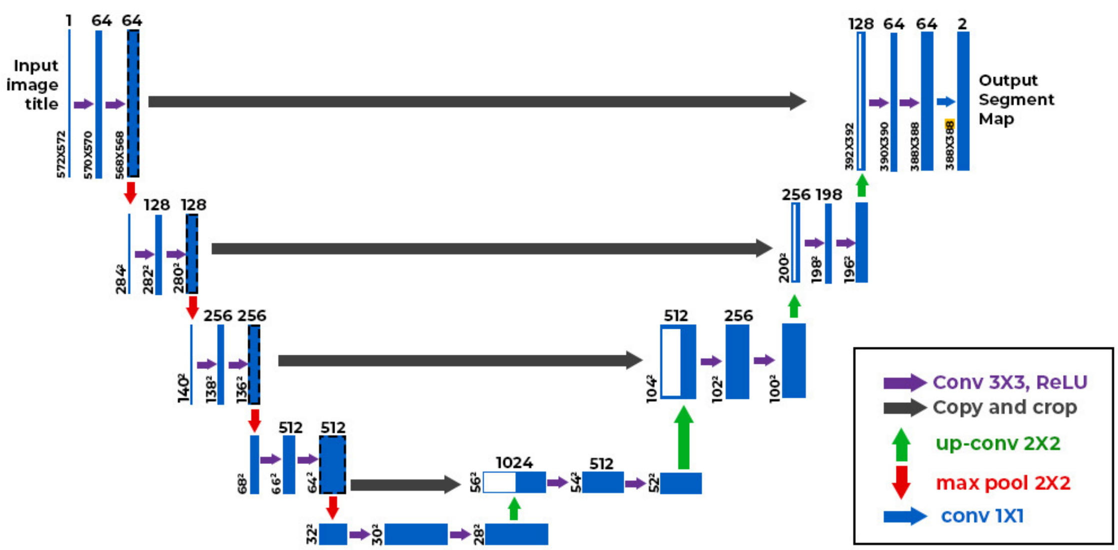

Figure 3: U-Net Architecture

The name U-Net takes place because of its U form. It consists of an autoencoder where the extraction of features is done, and the second part which is the decoder who reconstructs the input image using the latent space (obtained by the encoder), and finally a softmax function is applied to every pixel to classify them in each of the three classes. In the encoder, or more commonly known as backbone, as said, is the part of the network to identify thoroughly the image. Some of the most common are the following. ResNet50, MobileNet, EfficientNet or Xception among others. Due to the fact that this algorithm is going to be runned in the Jetson Nano, MobileNet was chosen as the best option because of using less memory, less time inference and less energy consumption. More specifically it was used the **MobilenetV2**, which is the most updated.

4.2 Training Model (PC)

Once we have the architecture is essential to train the model. Although the selected model should be feasible to train in the Jetson Nano device, it has been trained in an auxiliar PC to reduce as much as possible the training time, and the inference part is done in the Edge Device.
In the dataset [6], there are 60 images with their corresponding color masks, however, only using this few images is difficult to achieve great results, so there are to main options:

- **Data Augmentation**: technique that consists of rotating, scaling, changing textures amongst in order to have more images.
- **Fine-Tuning**: Using pre-trained weights acquired by training CNN with huge amount of images in order to detect in a good manner the essential features. In this case, the data set used was ImageNet, which contains a huge amount of plant images.

For this model, the fine-tuning selection was adopted because of its facility and the security of good results. Having defined all the architecture and training method, it is time to train the algorithm.
The paper show different hyperaprams taken into account to train the model. The batch size consists of the number of images as packs to the network to update the weights, so the selected number was 2 as the best option. Taking into account how less value, the generalization is better, and its capabilities to process is higher (so the model can also be trained in the Jetson Nano). For other hyperparameters such as **learning rate (0.0001)**, **momentum (0.9)**, and **epsilon (1e-8)**, the GridSearch algorithm was applied over some ranges of values to try different combinations. These hyperparameters give us the best results. Also, the final method was to train first the decoder part and then later train the encoder and the decoder part at the same time. In that way, it is ensure that the encoder of the backbone does not forget the previous features of million of images.
The **input image** consists of the simulation of the picture taken by the camera. An example will be as follows.

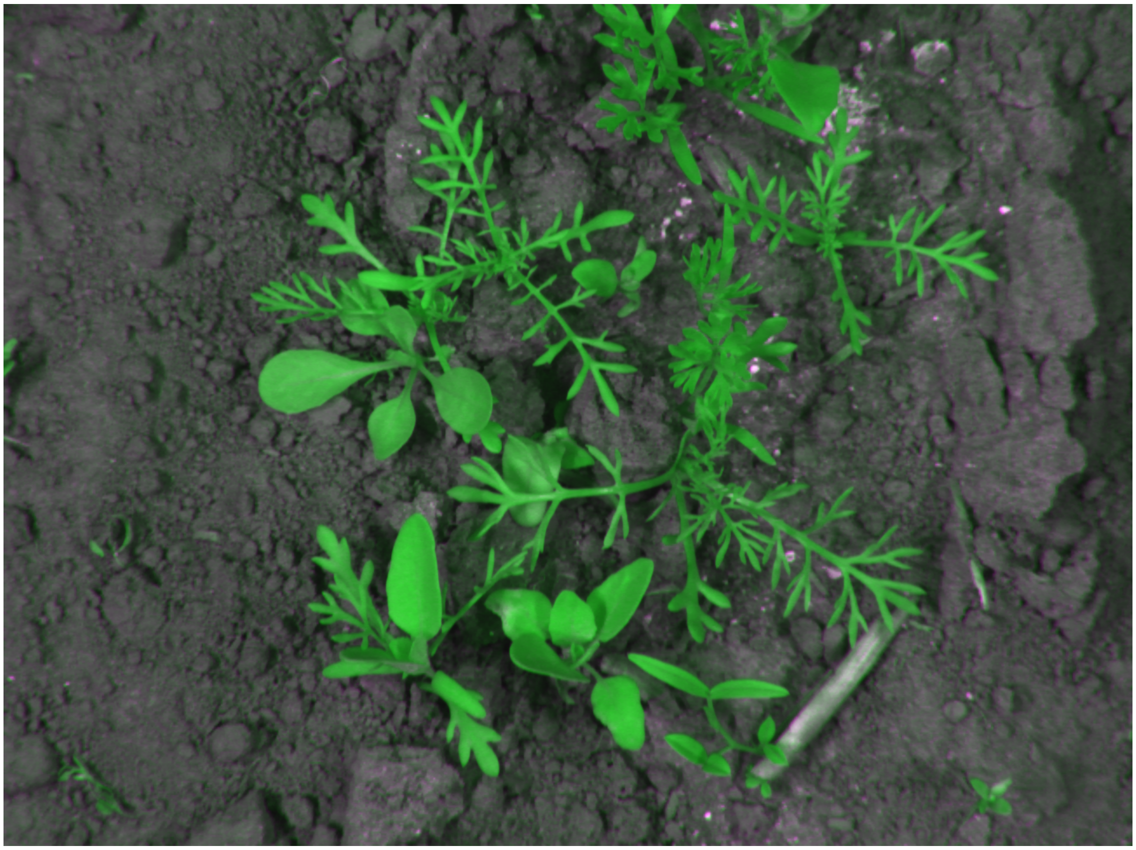

Figure 4: Input Image

Now the algorithm has the input it should predict the health part and the weed using colors. The first image below represents how the image should be labeled and the second one the predicted form the algorithm. Note that the prediction has used other percentage of mask so that there is no mistake in differentiating real and predicted labels.

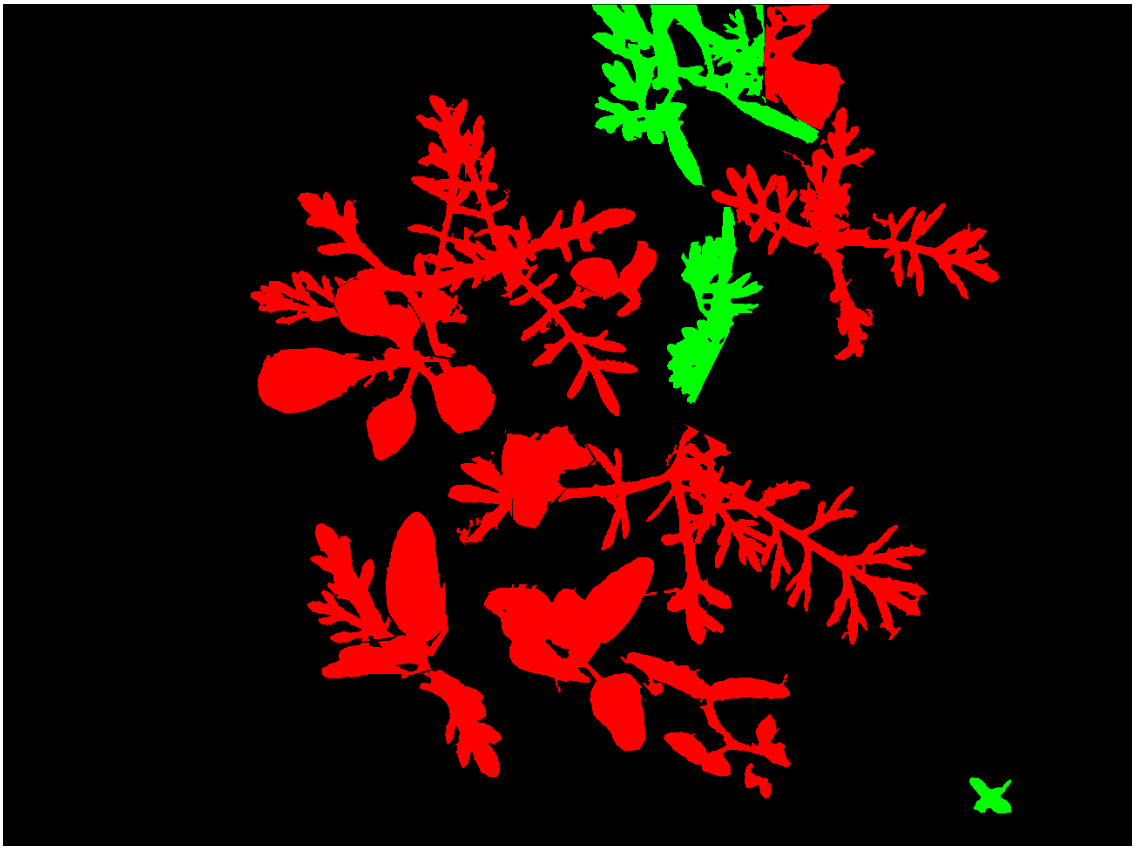

Figure 5: Real Label

Note that the colors represent the three possible classes which are black (background), green (healthy part of the carrot), and red (unhealthy part or weed), where the herbicide should be sprayed.

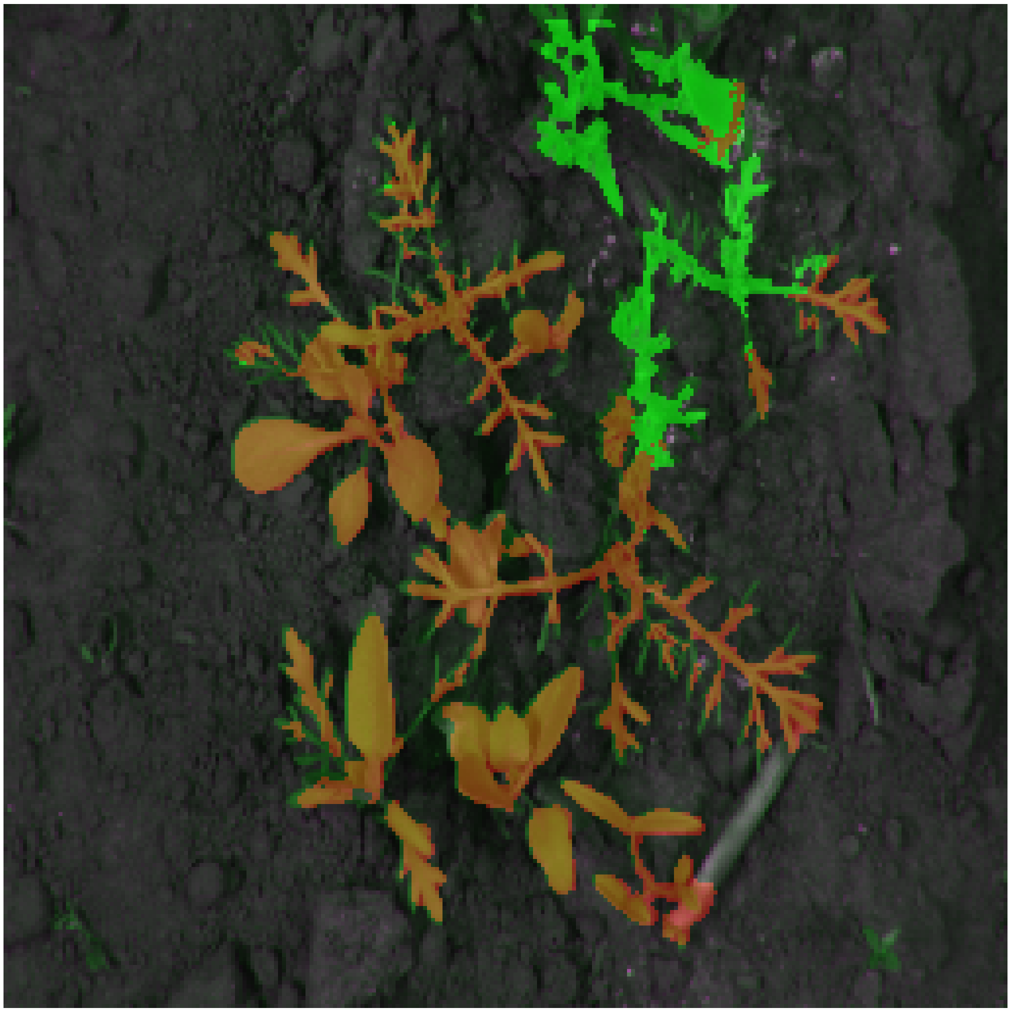

Figure 6: Predicted Label

This shows a prediction done in the PC, now this inference will be done inside the Jetson Nano and compare how it is done. To make it possible, it is necessary to save the model in order to be able to load all the weights and layers after training.

4.3 Validation (Jetson Nano)

Loading the Unet.h5 model saved it's time to use it in order to infer it. Taking into account the programming model used **Tensorflow** which consists of an open source library used to train machine learning and deep learning models. This library is supported by Jetson Nano, instead Nvidia cretated **TensorRT** which is an optimization of Tensorflow. Using this optimization, the inference time is reduced due to the possibility of reducing the bits precision (in this case is the same as the PC because it works well and classifying pixels requires certain precision - float32), optimizing operational order and merging some of the convolutional layers in one.

Now, by doing the prediction in the Jetson Nano and visualizing the result, we can see that the algorithm seems to work properly.

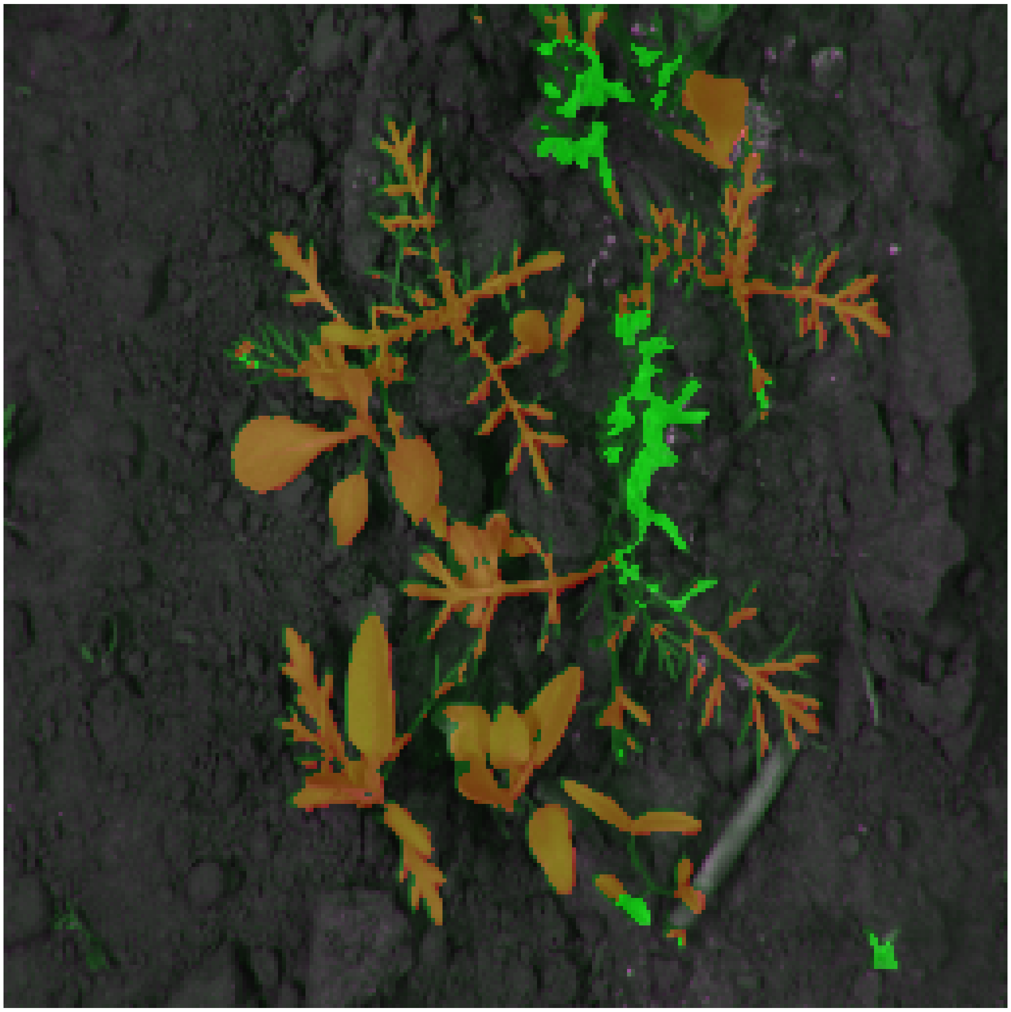

Figure 7: Predicted Label Jetson

4.4 Results

In addition to visualizing the results, it is important to measure them numerically. The first idea was to study an indicator such as the accuracy which counts the proportion of the total pixels well classified over the total pixels. In the very first implementation, the **accuracy** shows **98%**, however noticing that most of the photo is background is in advance easy, so instead the **mIOU** (mean Intersection over Union) that ponders the number of pixels well classifies knowing the total of each class.

$$mIoU = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i + FN_i}$$

The results given in the paper showed the better results using DeepLab using where 72%, however, these U-Net models show a result of **70%** and noticing that the model is faster, stable, and scalable. In conclusion, the application of this AI algorithm solves the problem. The agriculture of precision takes a difficult use of case to solve, and taking into account there are 3 classes and not 2, it is important to remark that this value of the metric shows a great performance. Fruit segmentation in natural orchards [7] acquires 86% mIoU, however, the task was only to detect fruits that are not as difficult to detect as healthy or unhealthy crops, taking into account that searching for these features is quite complex.

5 Execute the model

In order to run, the model on the Jetson Nano is essential to know the root file directory.

```
Desktop/
 CWFID/
     dataset/
     model/
     saved model/
     Trt_model.py
     Unet_TRT.py
     run_unetRT.sh
```

The dataset folder contains the input images and color masks, however, inference is only affected by the input images. The model folder contains the.h5 and the saved model the necessary to load the tensorRT. In case the saved model not exists, please run Trt_model.py.
Lastly, to run the inference of an input image, just on the terminal type: `.\run\unetRT.py`

6 References

[1] Md. Waquar Azam, ”Agricultural Crops Image Classification,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/mdwaquarazam/agricultural-crops-image-classification/data. [Accessed Apr. 2, 2025].

[2] Ram Krishna, ”Corn Leaf Infection Dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/qramkrishna/corn-leaf-infection-dataset. [Accessed Apr. 3, 2025].

[3] M. Padhiary, D. Saha, R. Kumar, L. Nayaran Sethi, A. Kumar, ”Enhancing precision agriculture: A comprehensive review of machine learning and AI vision applications in all-terrain vehicle for farm automation”, Elsevier, vol. 8, 2024 .

[4] A. Ribeiro Seijas, J.M. Bengochea-Guevara, M. Todeschini, D. Andújar, Late Botrytis TERRAS 2021, Digital CSIC, Nov. 2, 2021. [Online], Available: https://digital.csic.es/handle/10261/253427 [Accessed Mar. 27, 2025]

[5] S. Álvarez Fernández, D. Prieto López, A. Ribeiro, ”D4.1 Pilot 1 objectives, requirements and design”, Cordis - EU, Dec. 28, 2023 [Online]. Available: https://cordis.europa.eu/project/id/101017111/results [Accessed Mar. 25, 2025]

[6] E. Assunçao, P.D. Gaspar, R. Mesquita, M.P. Simões, Kha. Alibabaei, A. Veiros, H. Procença,Real-TimeWeed Control Application Using a Jetson Nano Edge Device and a Spray Mechanism , Sebastopol, CA: MIP, 2022. [E-book] Available: MPI.

[7] H. Kang and X. Wang, ”Semantic Segmentation of Fruits on Multi-sensor Fused Data in Natural Orchards,” Cornell University, Aug 4, 2022. [Online]. Available https://arxiv.org/abs/2208.02483. [Accessed Apr. 26, 2025]