Automated Fruit Detection and Counting for Precision Agriculture: A Deep Learning Approach for 2D High-Resolution Orchard Images

Francesco Zambelli
Pietro Miglioranza
Gaetano Ricucci

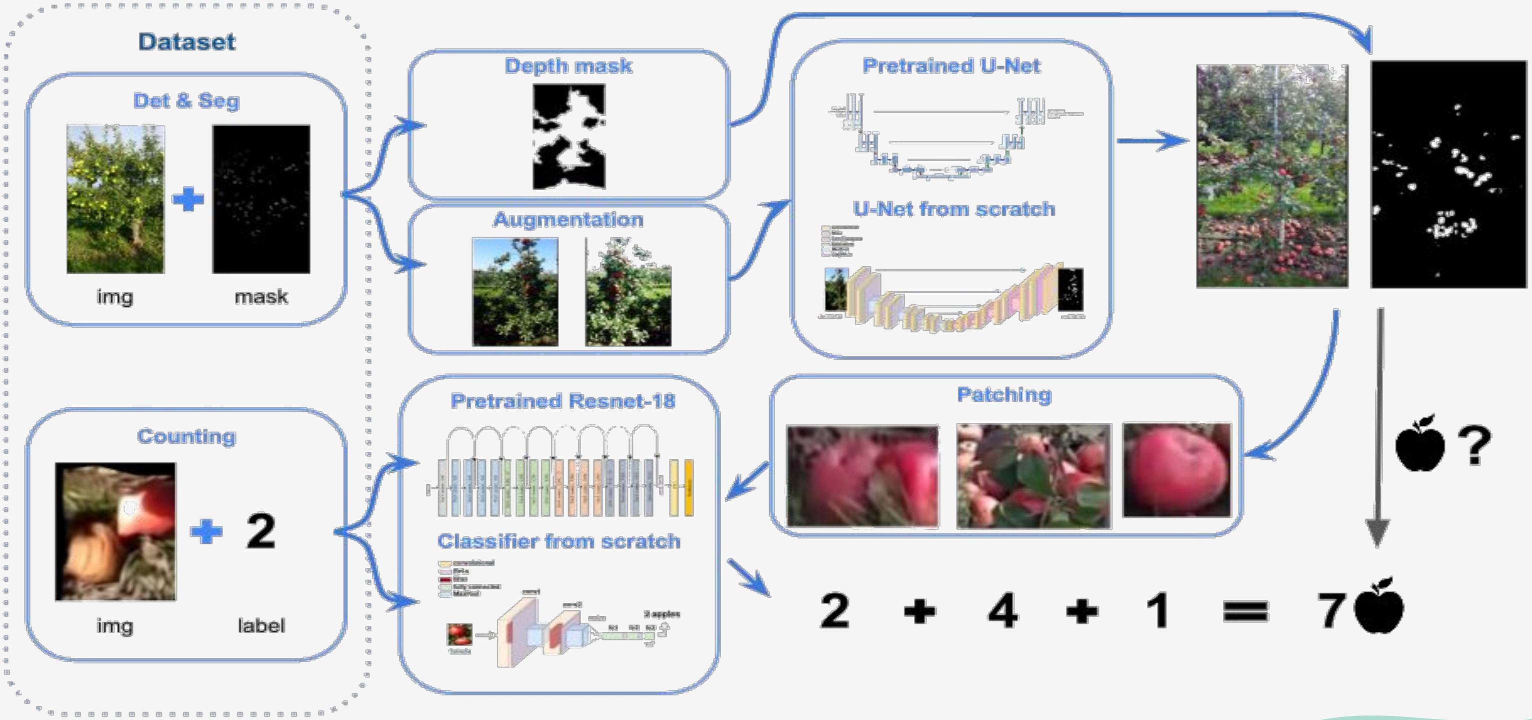
PROBLEM & SOLUTION

Apples manual counting is time consuming and often leads to erroneous orchards yields estimation

Build an algorithm that, starting from 2d apple trees images, automatically recognize and count apples

Combining the counts of each tree give an estimate of the total harvest

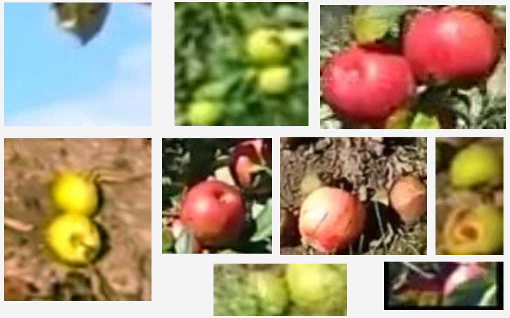
GENERAL PIPELINE



DATASET

COUNTING

Train Samples



Test Samples

The Counting dataset is composed of image patches of different dimensions .

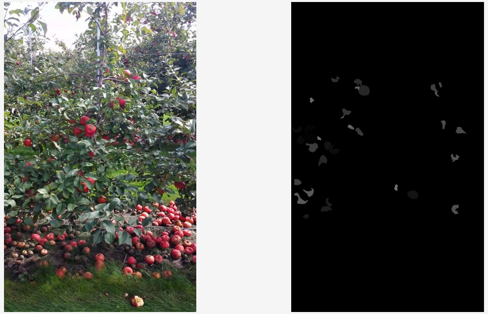
- Training Set:
  - 64595 patches of apples captured under different conditions
- Validation Set:
  - 3395 patches of apples captured under different conditions
- Test Set:
  - 2875 patches of apples extracted from trees
- Pre-Processing:
  - Padding to a square
  - Resize to 64x64 pixels

DATASET

SEGMENTATION

Train Sample

Test Sample

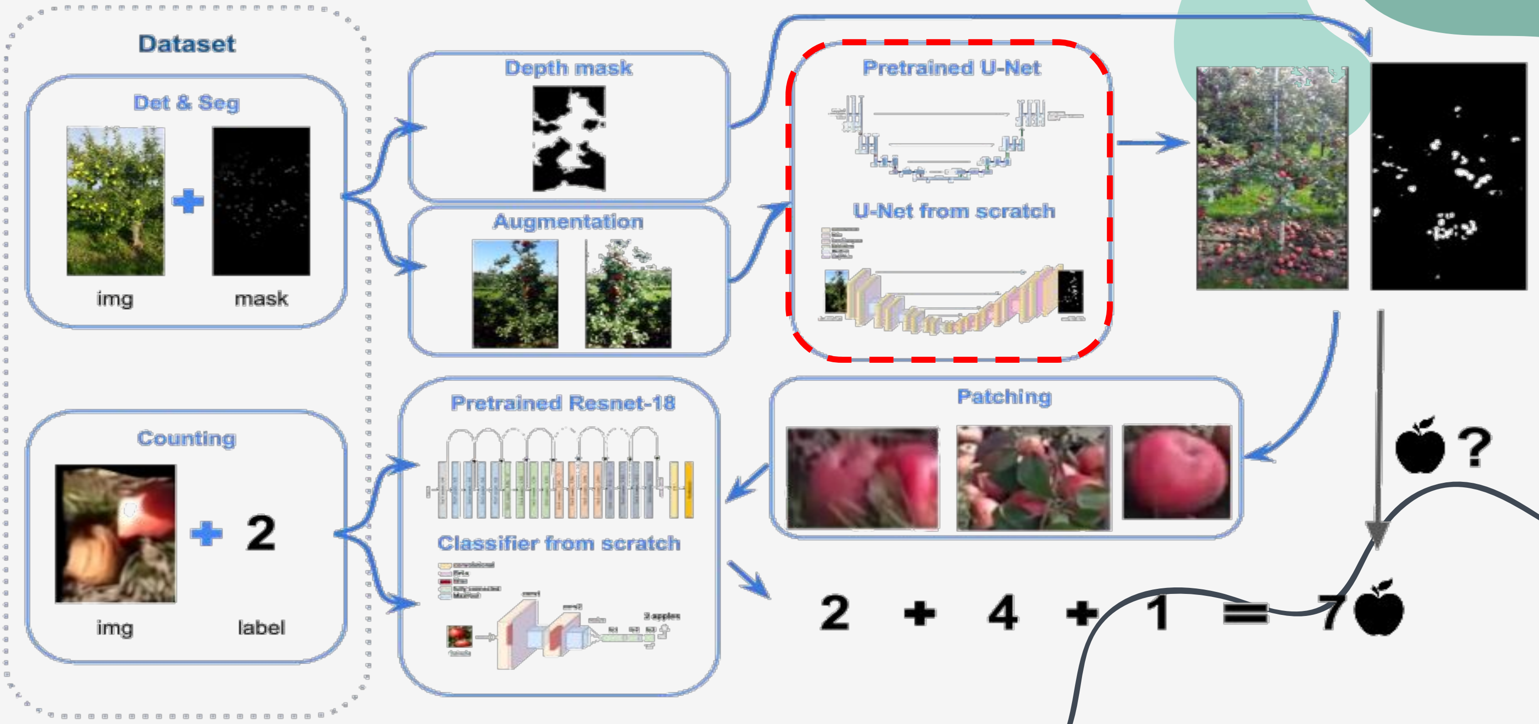


The Segmentation dataset is composed of High resolutions images (1280x720) extracted from a video recording.

- Training Set:
  - 603 images with corresponding mask
- Validation Set:
  - 67 images with corresponding mask
- Test Set:
  - 370 images with corresponding mask

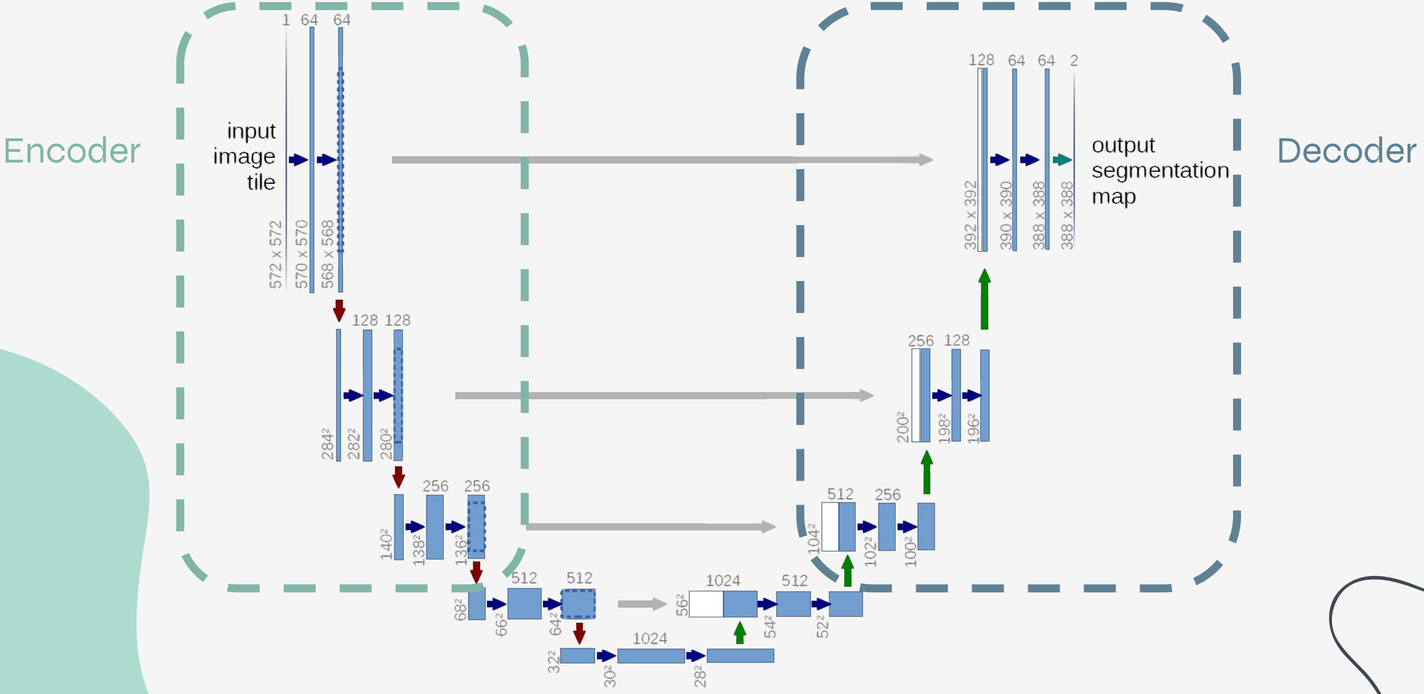
- Data Augmentation:
  - Random brightness adjustment(0.7-1.3)
  - Random horizontal flip

SEGMENTATION



SEGMENTATION

Approach : UNet

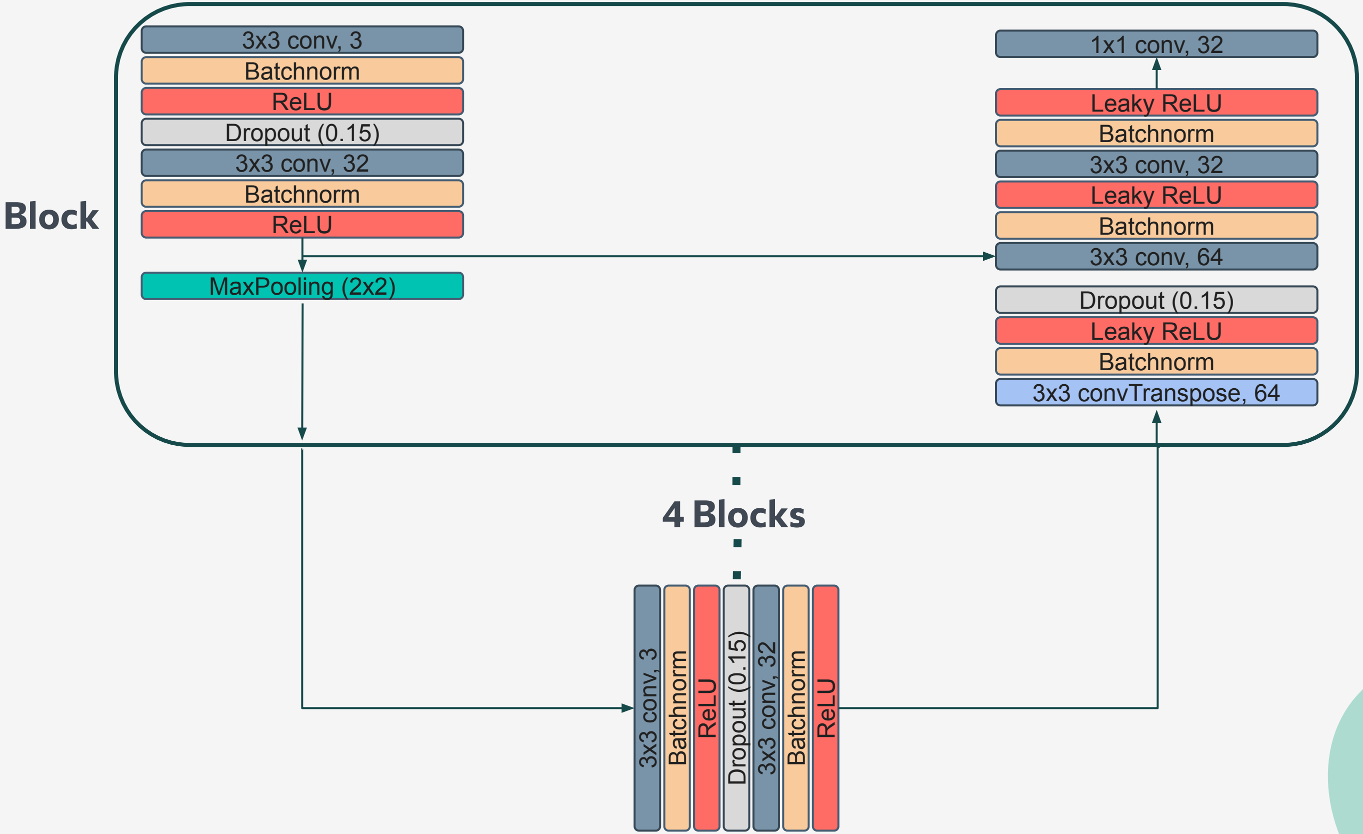


SEGMENTATION
Architectures

1. Custom implementation
   a. Trained for 100 epochs
   b. Adam Optimizer with l.r = 0.0005
   c. Binary Cross Entropy Loss with weights.
2. Pre Trained architecture on ImageNet dataset
   a. Fine-Tuned for 100 epochs
   b. Adam Optimizer with l.r = 0.0005
   c. Binary Cross Entropy Loss with weights.

SEGMENTATION

1) Custom UNet



RESULTS

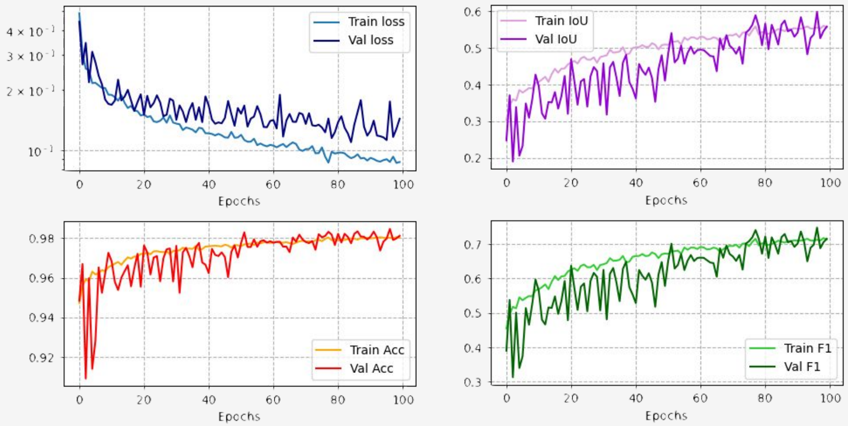
1) Custom UNet

Validation

Ground Truth	Segmentation
	

Test

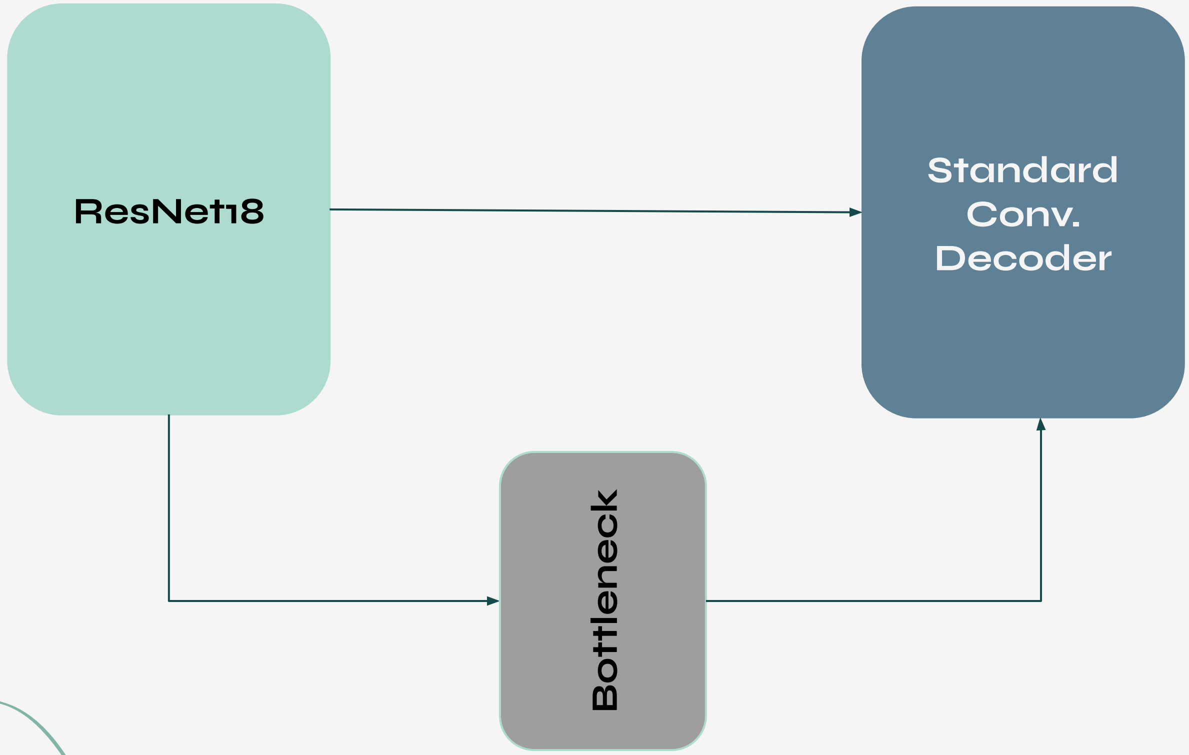
Ground Truth	Segmentation
	



Custom	Normal	Augmented
Val IoU	0.506	0.551
Test IoU	0.394	0.416

SEGMENTATION

2) Pre-Trained UNet



RESULTS

2) Pre-Trained UNet

Validation

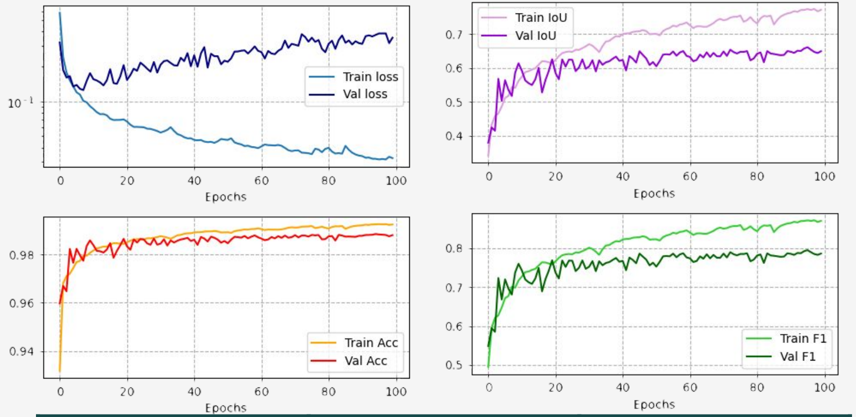
Ground Truth

Segmentation

Test

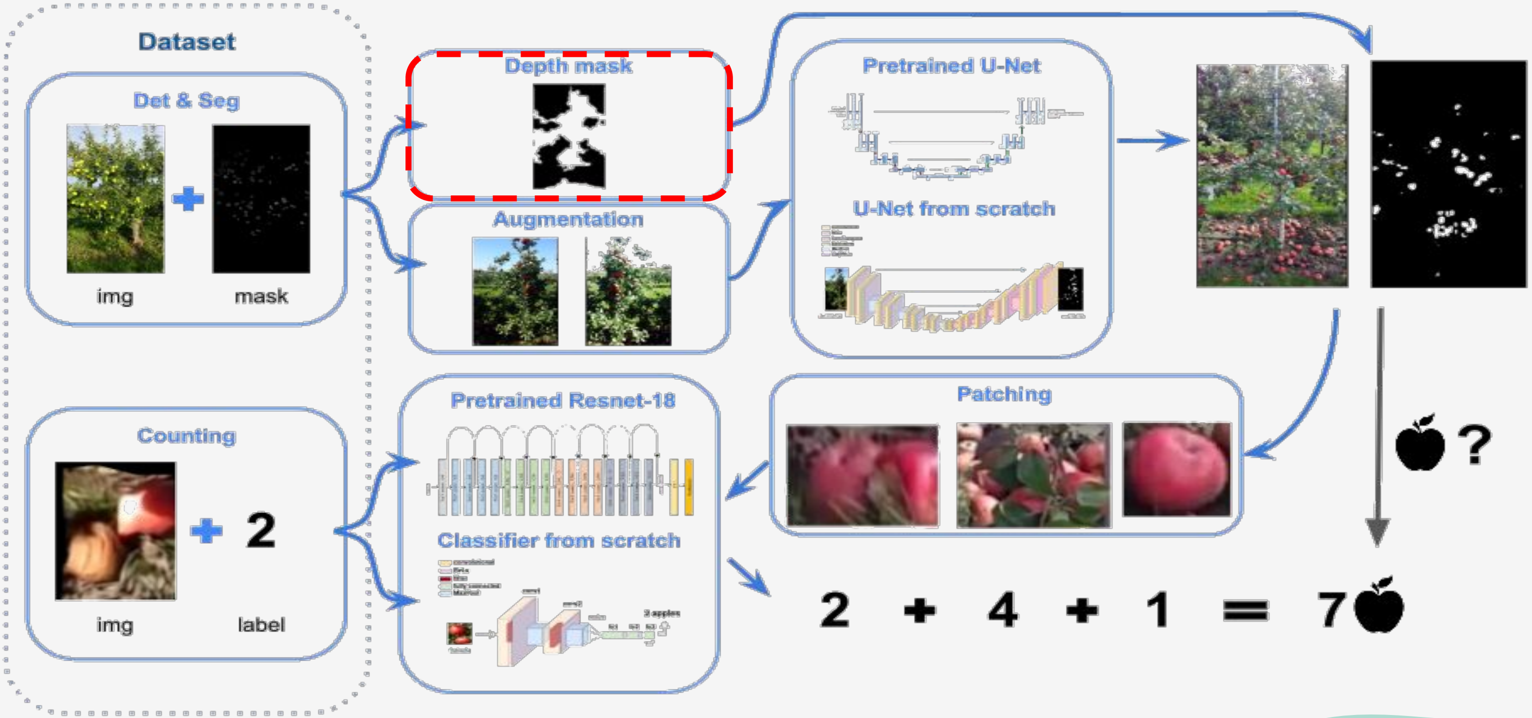
Ground Truth

Segmentation



ResNet	Normal	Augmented
Val IoU	**0.665**	**0.640**
Test IoU	**0.559**	**0.549**

DEPTH ESTIMATION



DEPTH ESTIMATION

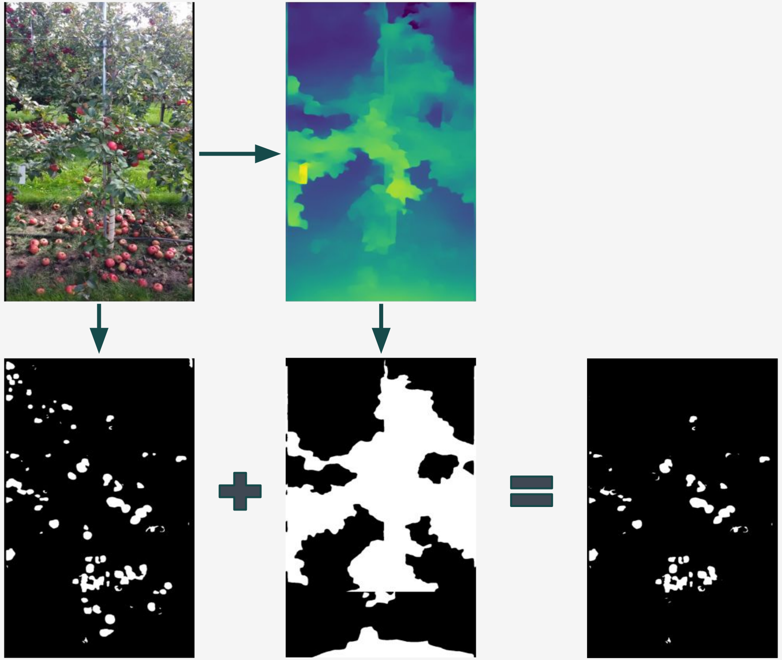
Problem: many apples that are not on a tree are detected

Ground truth

Our segmentation

DEPTH ESTIMATION

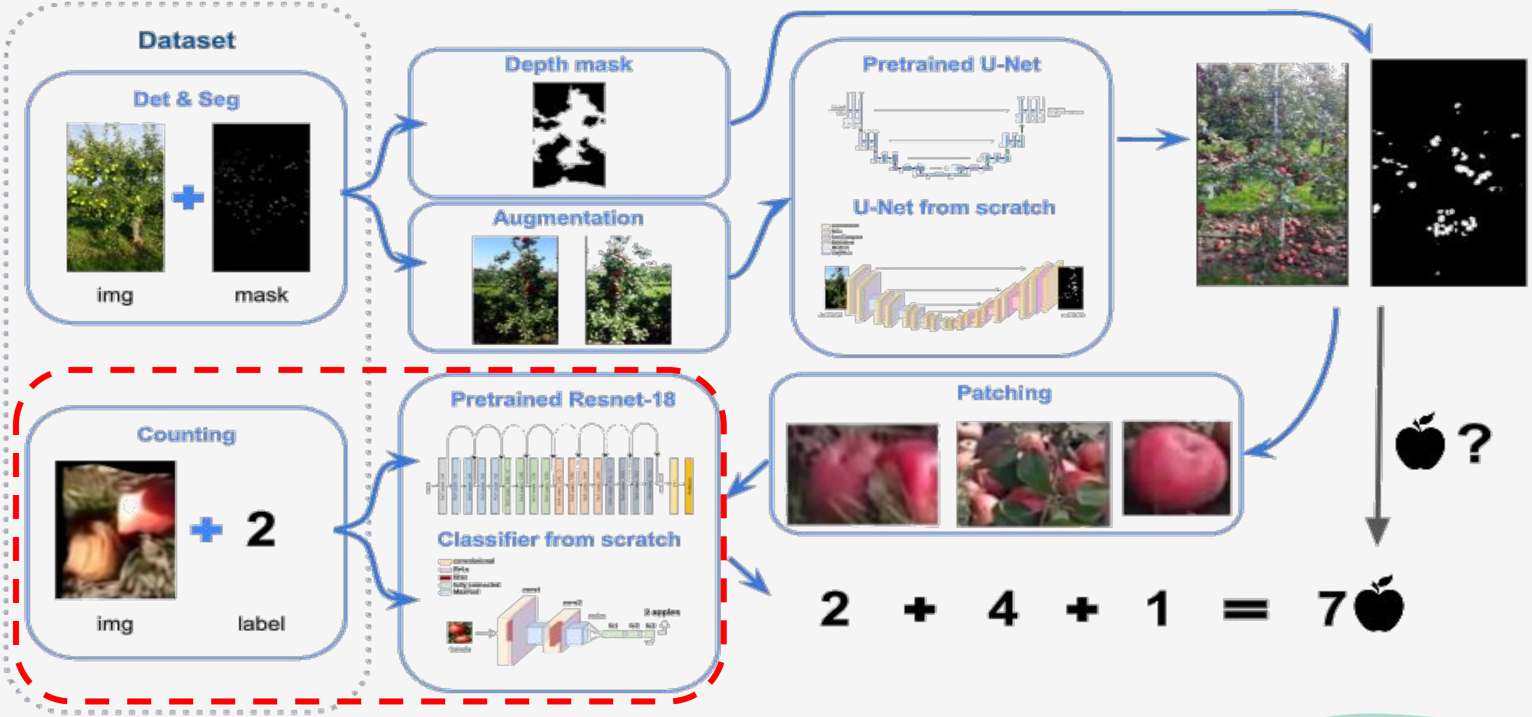
Solution : **try to filter selected apple using the mask of the tree**



- Pre-trained MiDas v3:
  - A Res-Net based architecture trained on different dataset used for computes relative inverse depth from a single image
- Threshold:
  - An empirical threshold is set on the inverse depth to exclude ground.

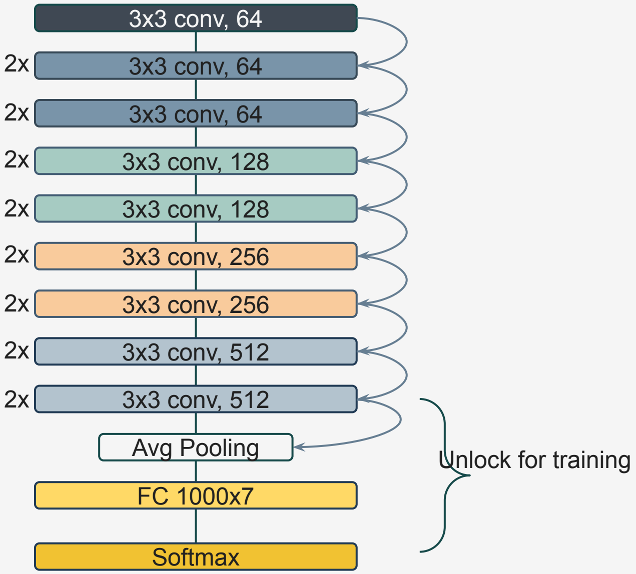
Results : **detected apples are significantly less than before but still too many compared to ground truth**

COUNTING

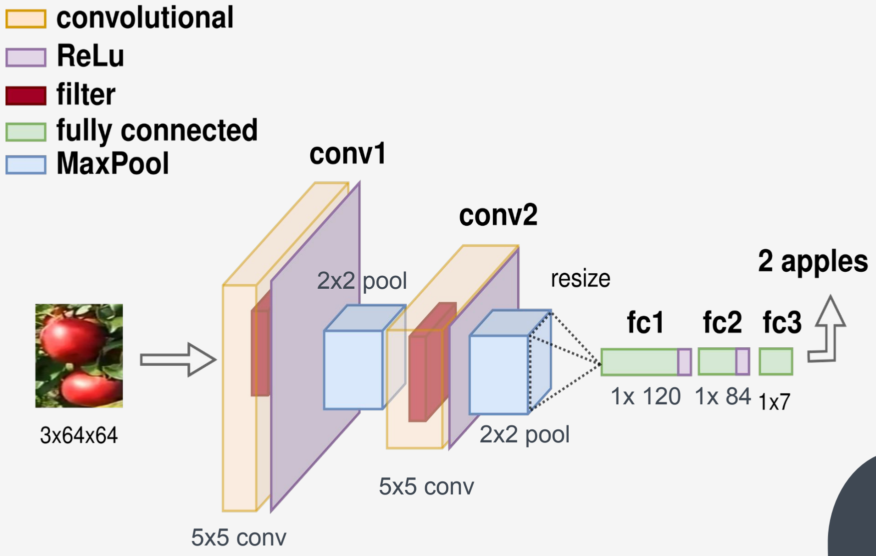

Dataset
Det & Seg
img
mask
Depth mask
Augmentation
Pretrained U-Net
U-Net from scratch
Counting
img
2
label
Pretrained Resnet-18
Classifier from scratch
Patching
?
2 + 4 + 1 = 7

COUNTING

Pretrained Resnet18



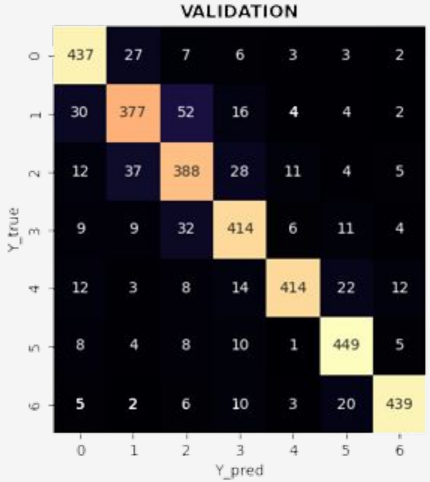
Simple CNN

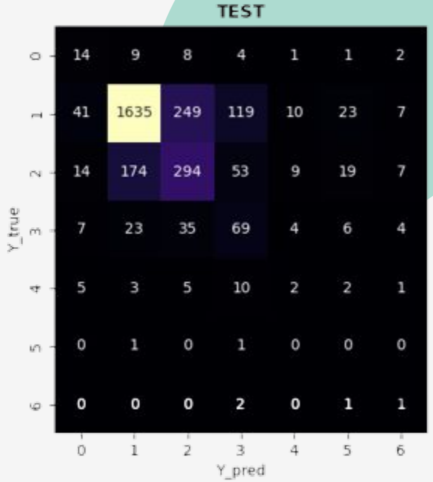


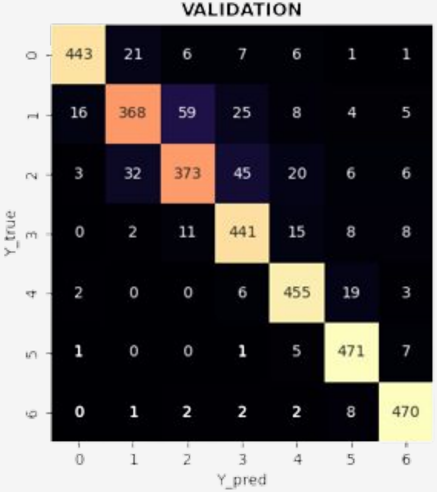
Results

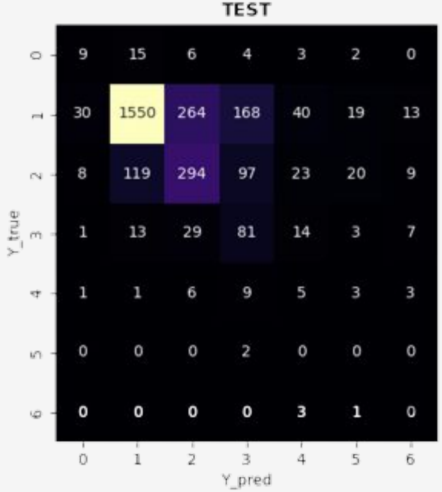
ResNet	Val	Test
Accuracy	**0.859**	**0.701**
F1 score	**0.860**	**0.295**

Custom	Val	Test
Accuracy	**0.890**	**0.674**
F1 score	**0.888**	**0.276**





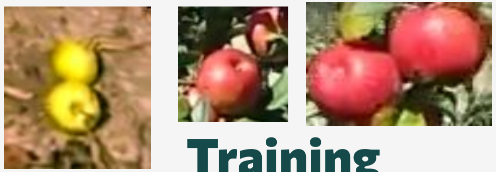




COUNTING

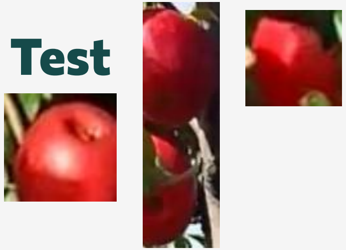
Problem: accuracy on the test set is low.

Possible reason: training images are probably hand-cropped, while the test images are cropped with a patching algorithm.



Training

Test

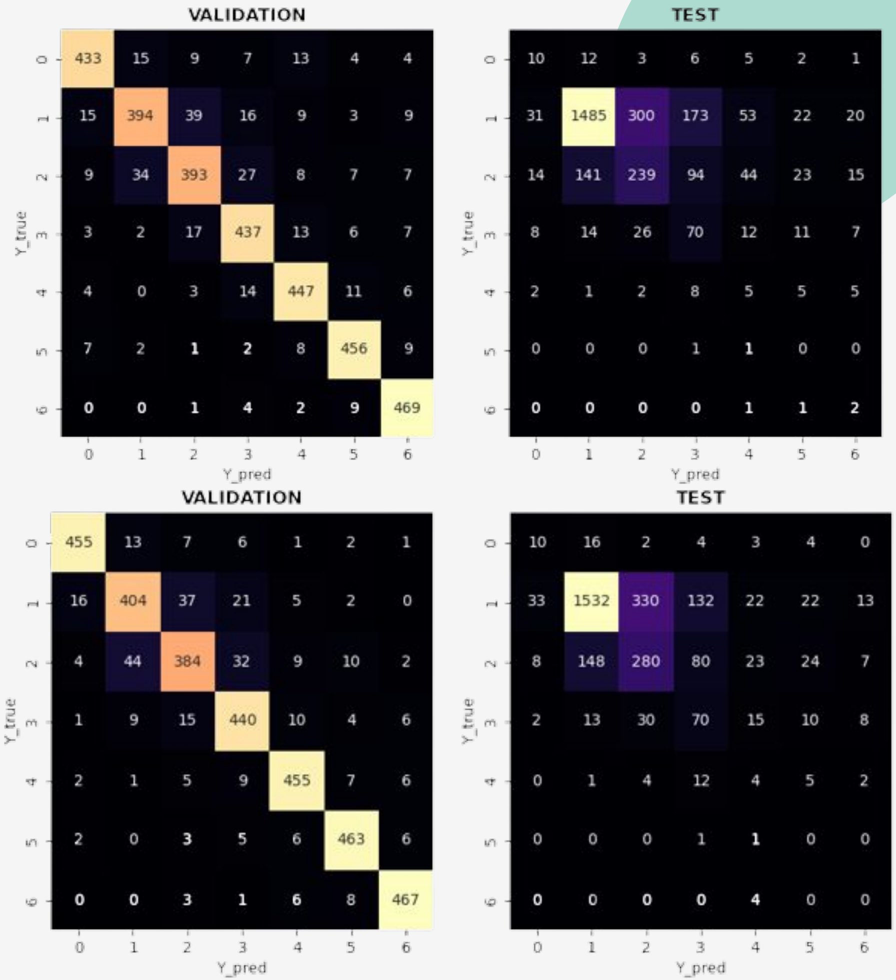


Solution: <u>randomly crop</u> training images using a scale of [0.8; 1] and then resize to 64x64.

Results with crop

Resnet	Val	Test	Val Crop	Test Crop
Accuracy	0.859	0.701	0.892	0.630
F1 score	0.860	0.295	0.892	0.261

Custom	Val	Test	Val Crop	Test Crop
Accuracy	0.890	0.674	0.904	0.659
F1 score	0.888	0.276	0.903	0.268

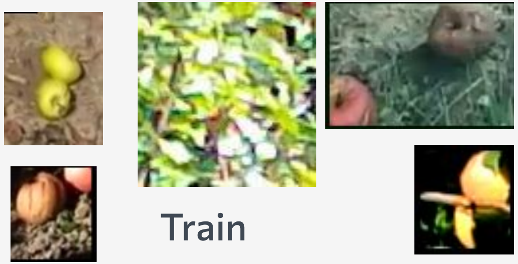


COUNTING

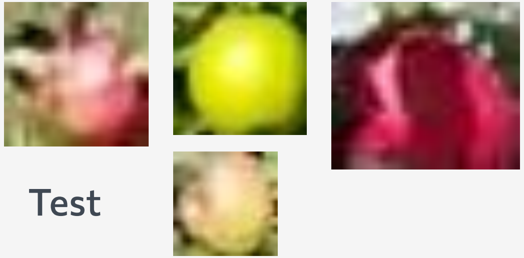
Problem: accuracy on the test set is low.

Other possible causes:

- **different background between train and test set;**

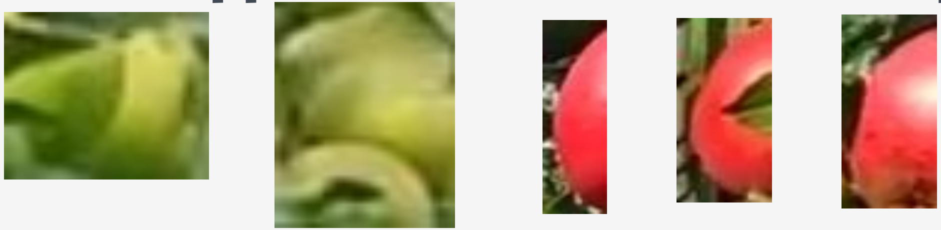


Train

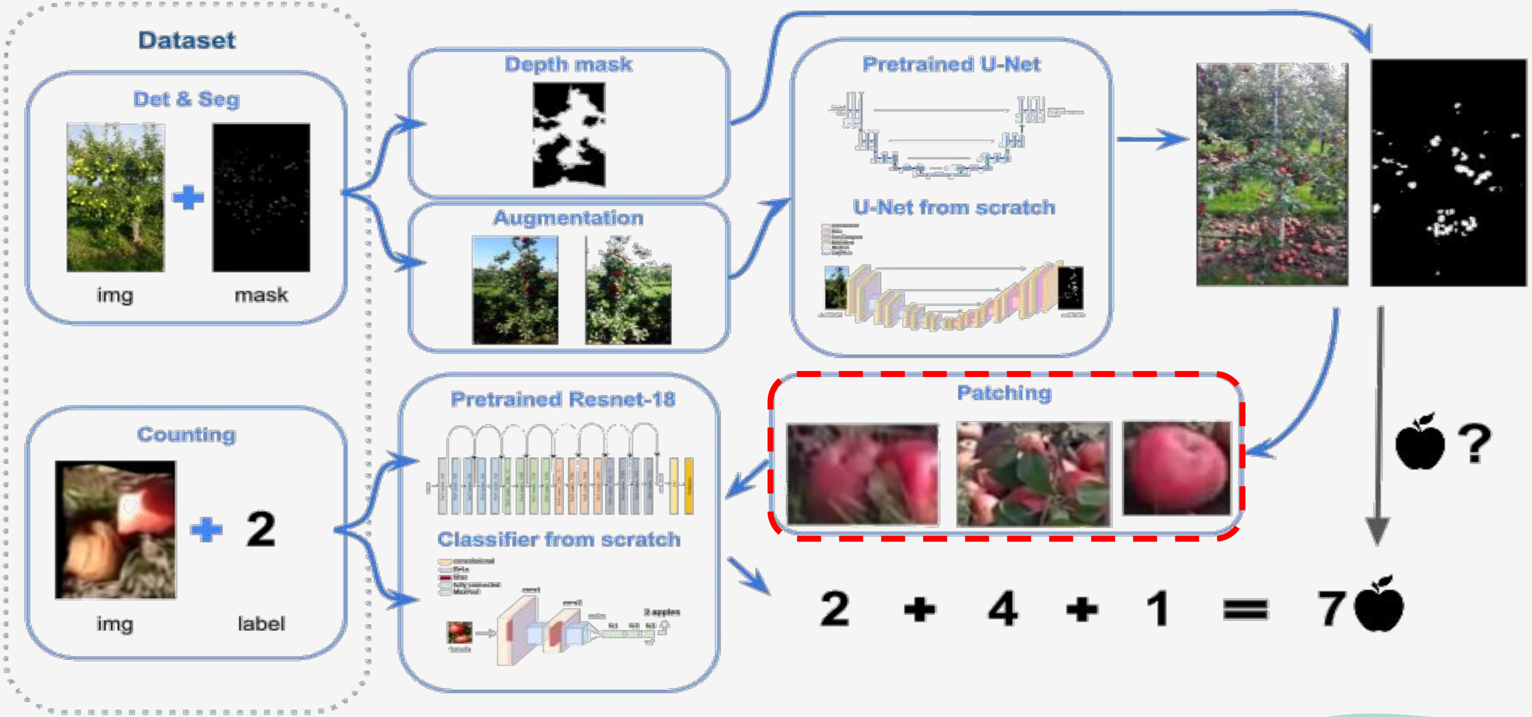


Test

- **some apples in train dataset are cropped a lot;**

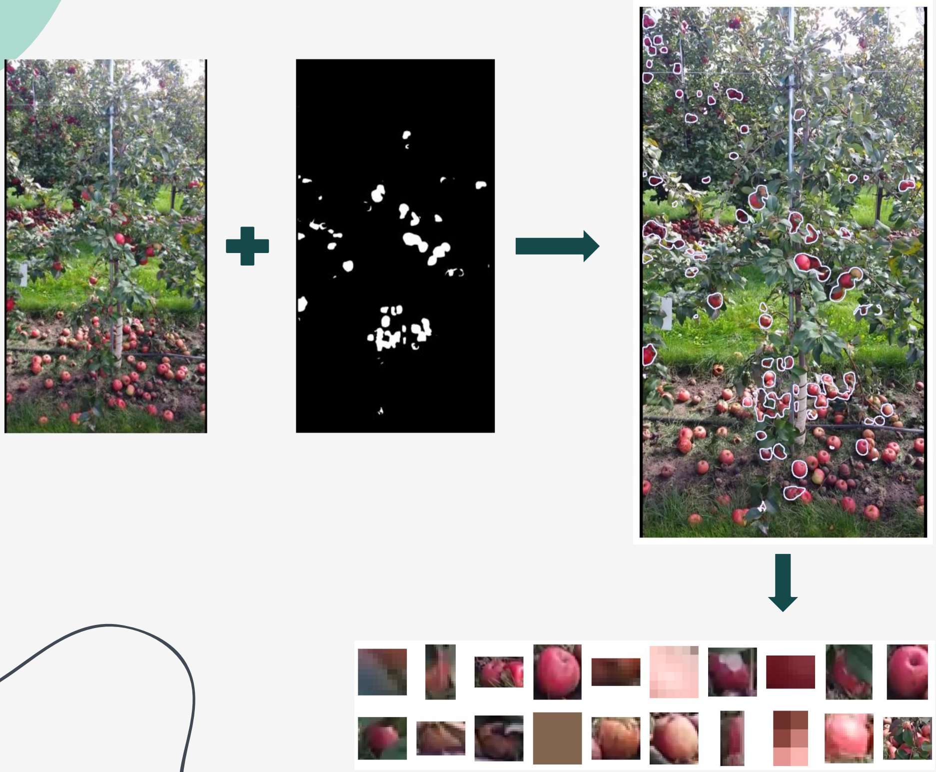


PATCHING



Dataset
Det & Seg
img
mask
Counting
2
img
label
Depth mask
Augmentation
Pretrained U-Net
U-Net from scratch
Pretrained Resnet-18
Classifier from scratch
Patching
?
2 + 4 + 1 = 7

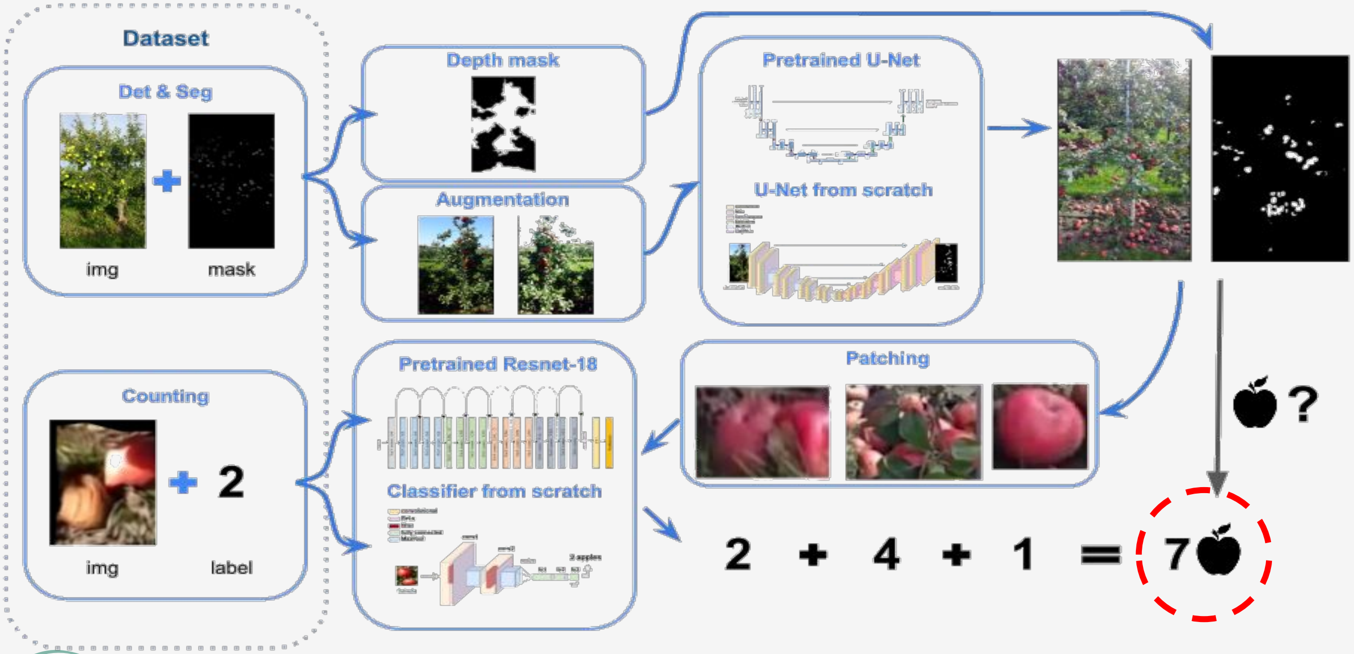
PATCHING



- Extract mask true segmentation
- Find all the connected areas
- Using openCV.findcontours to extract the bounding box of all the connected areas.
- Cut out the image around these boxes found to get the apple patches

COMBINING

Goal: by combining the results of the previous passages, we are now able to *estimate the total number of visible apples* for each tree picture



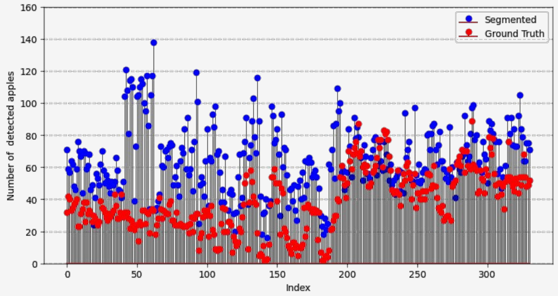
COMBINING

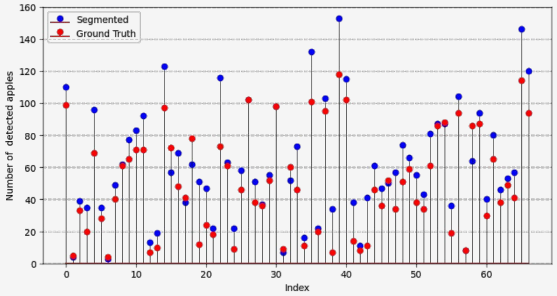
RESULTS

TEST

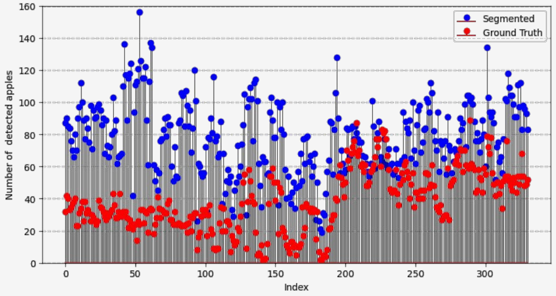
VALIDATION

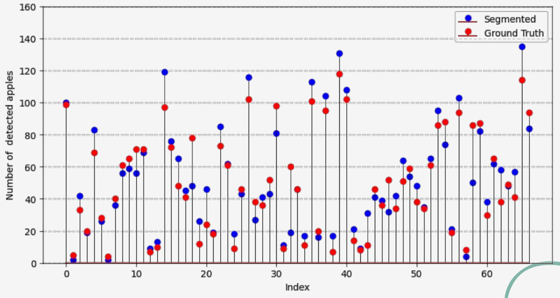
With depth-estimation filter





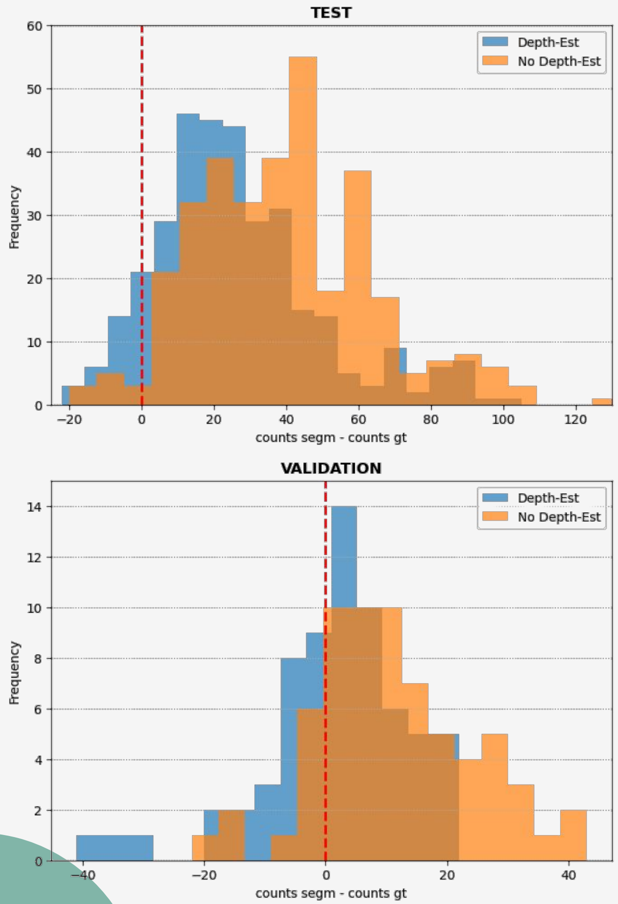
Without depth-estimation filter





COMBINING

RESULTS



The combined procedure to count the total number of apples leads to bad results:

- Number of apples detected from the segmentation mask are **systematically higher** than the the number extracted from ground-truth masks
  - This aspect highlights how the segmentation algorithm and the successive adjustments are **not enough** to select only apples present on the considered tree
- For the validation such an effect is **less evident** compared to the one in the test set
  - This suggests that images belonging to the test set are significantly different from the ones in training and validation set, or, on the other hand, that the trai set is not enough representative

OVERLAP MAXIMIZATION

Problem: many apples appear multiple times in images corresponding to successive frames of the video

Idea of solution:

- Align segmentation masks corresponding to successive images in a meaningful way
    - Keypoints matching on the original images
    - Overlap maximization directly on masks
- Select which patches of the two images correspond to the same apples and extract the corresponding count
- Subtract such a number from the sum of the total number of counted apples in the two images

OVERLAP MAXIMIZATION

Problem: many apples appear multiple times in images corresponding to successive frames of the video

Idea of solution:

- Align segmentation masks corresponding to successive images in a meaningful way
  - Keypoints matching on the original images
  - Overlap maximization directly on masks
- Select which patches of the two images correspond to the same apples and extract the corresponding count
- Subtract such a number from the sum of the total number of counted apples in the two images

01

Transformation

Apply a geometrical transformation composed by translation, rotation, scale to an image and measure the IoU with a second image

02

Bayesian Optimization

Explore the space of the possible transformation and find the parameters combination that maximizes the IoU

03

Images transformation

The best transformation is applied to the two images

04

Patches overlap threshold

Each patch of the first image is considered, and is computed the overlap with the secondo image. If it is greater than a thresh, the patch is considered to represent double counted apples

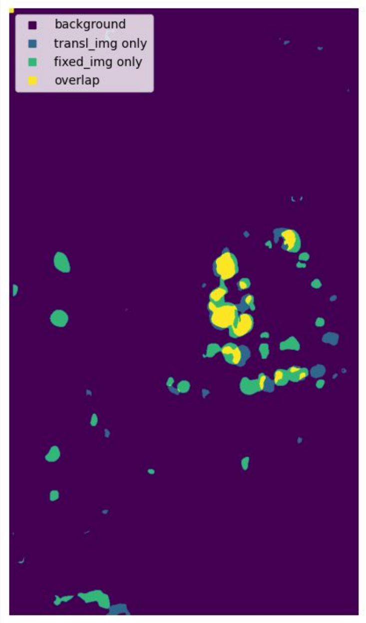
05

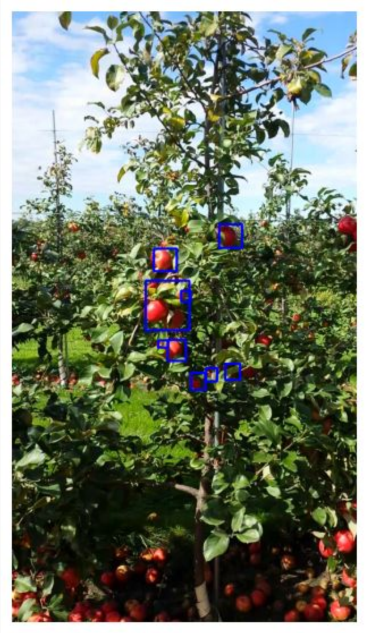
Count correction

Each patch detected in such a way is passed to the classifier, the values are summed and then subtracted to the sum of the total number of apples in the first and second image

OVERLAP MAXIMIZATION

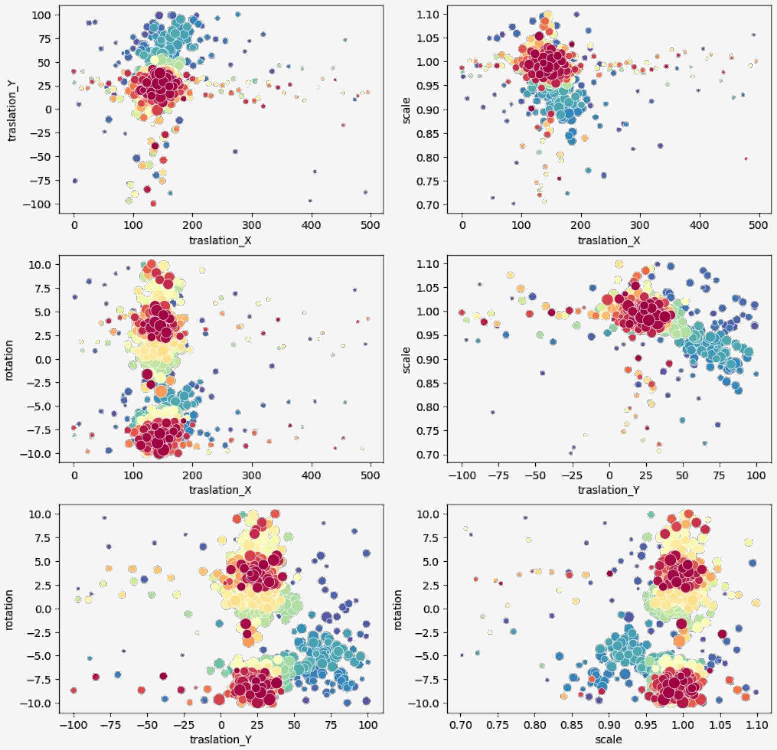
Results

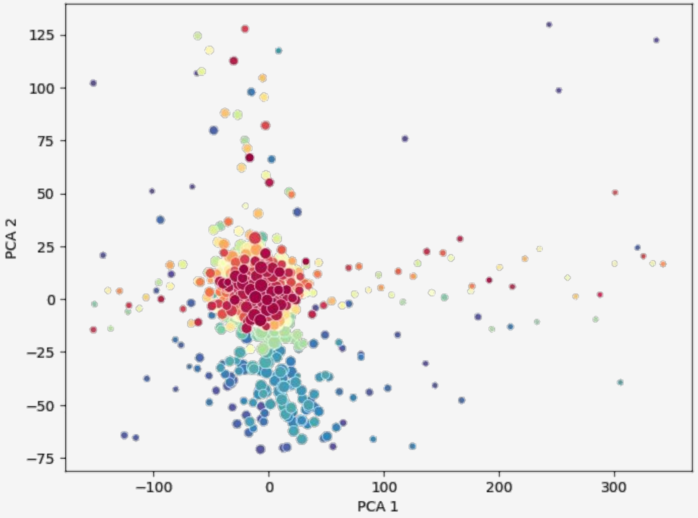




OVERLAP MAXIMIZATION

Parameter space Visualization





CONCLUSIONS and FUTURE WORKS

Bad results, overestimation of over 300%

⬇

- Apple detection via segmentation is a relatively easy task, but when dealing with yield estimation many more problems arise

⬇

3D reconstruction techniques are fundamental to reach good performances in a yield estimation task

- Deeper exploration of the depth estimation procedure could improve significantly the results even without 3D reconstruction

Thanks for the attention