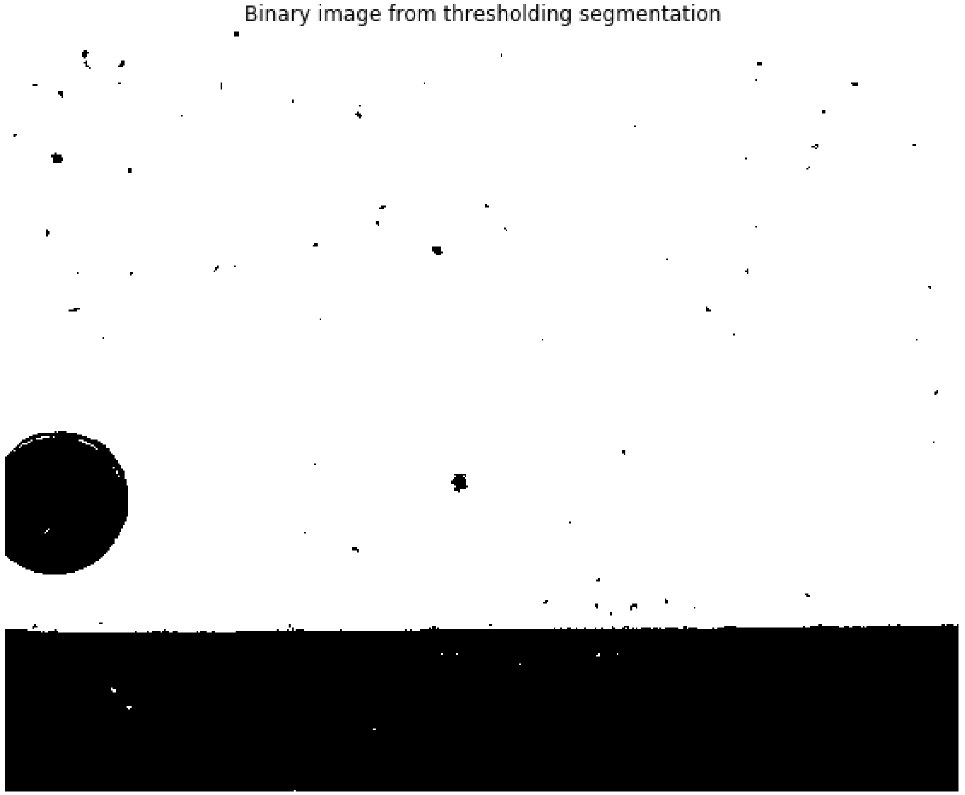
EPFL

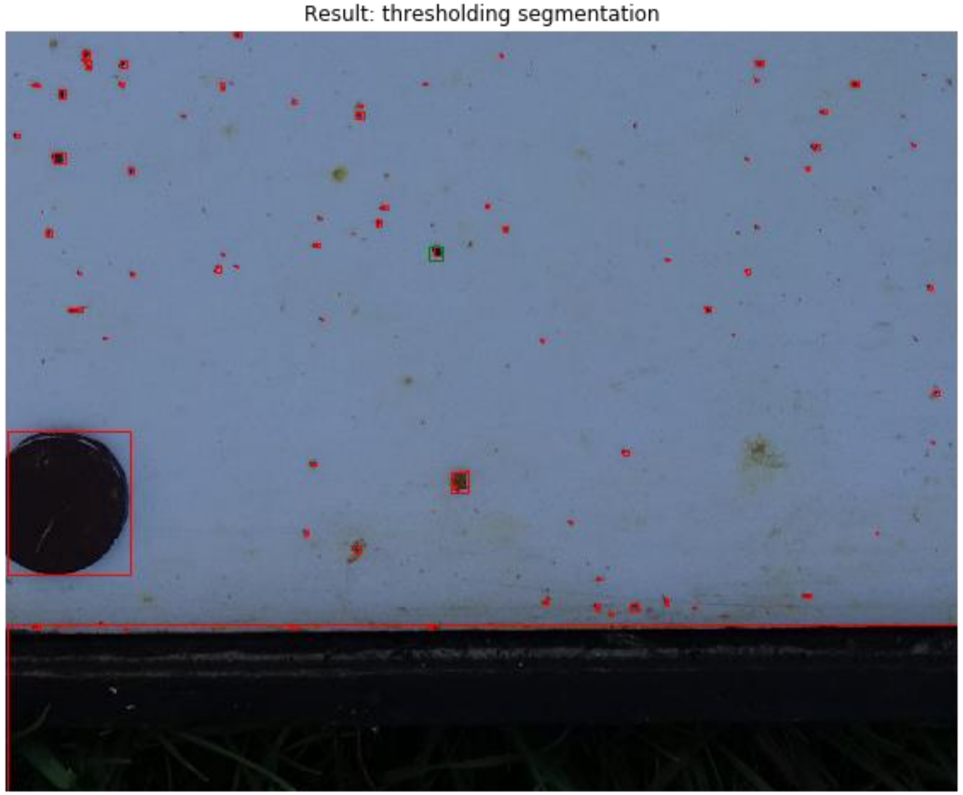
EE-451 Image analysis and pattern recognition

Special Project

Lorenzo Berardo, Xiao Zhou and Yi-Shiun Wu

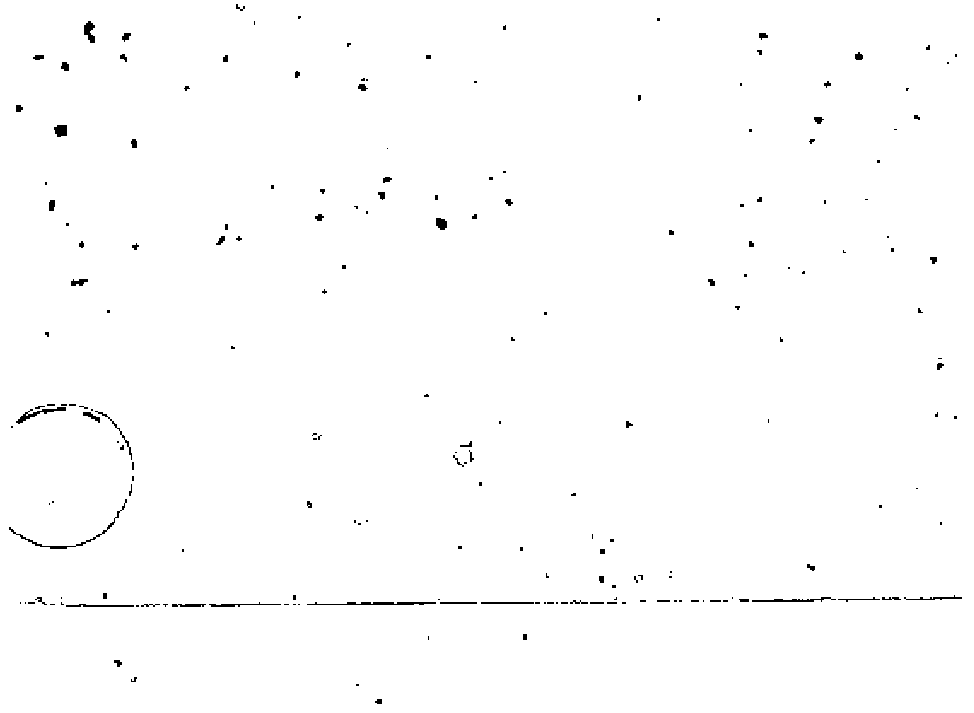
Segmentation with Thresholding


Binary image from thresholding segmentation

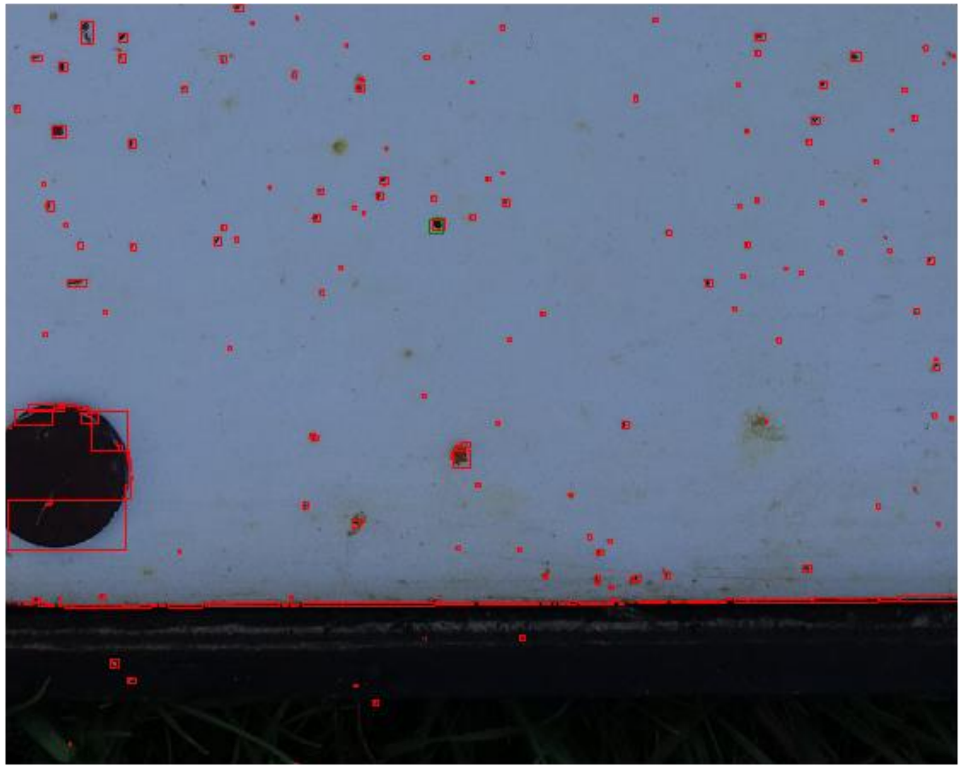

Result: thresholding segmentation

Segmentation with Edge detection

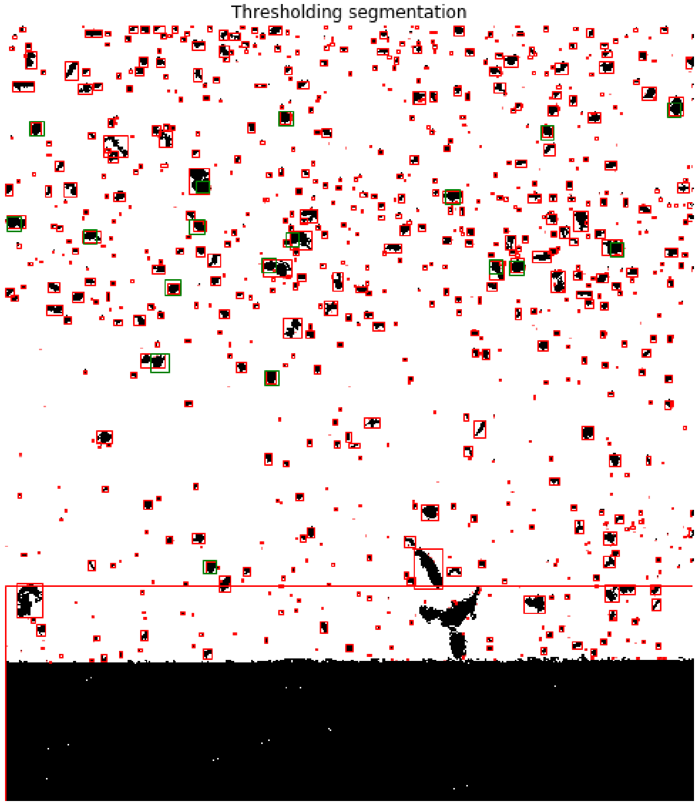
Binary image from edge segmentation

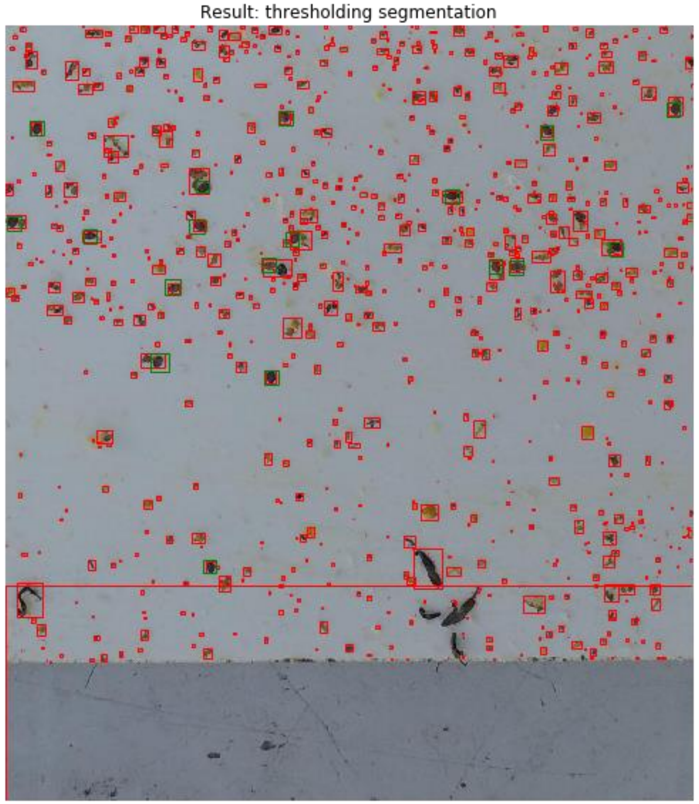


Result: segmentation with edge detection - Sobel filter

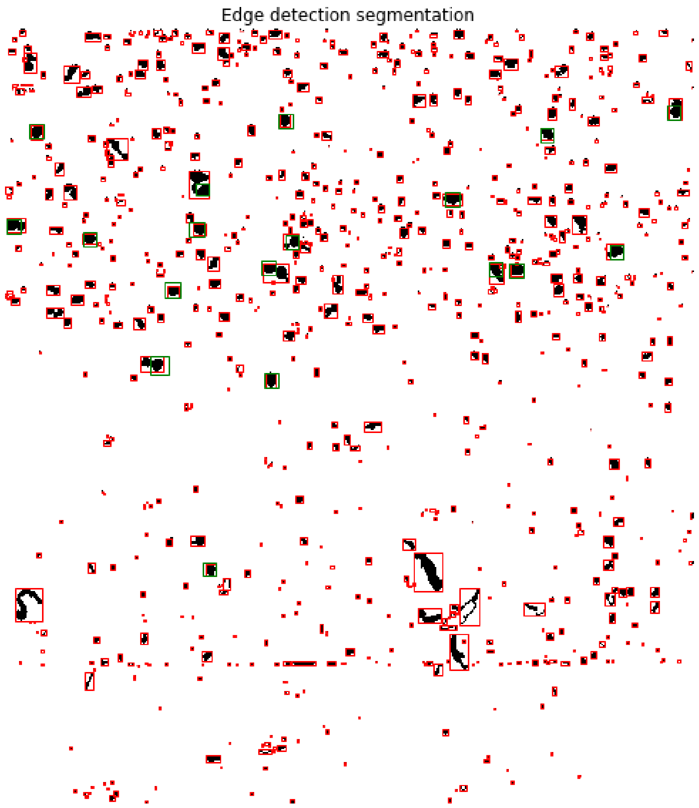


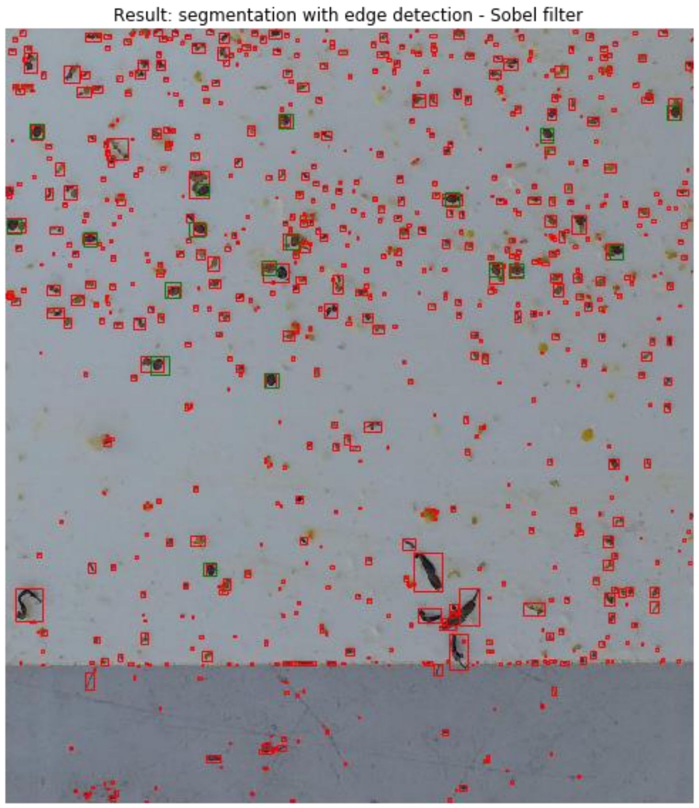
Segmentation with Thresholding



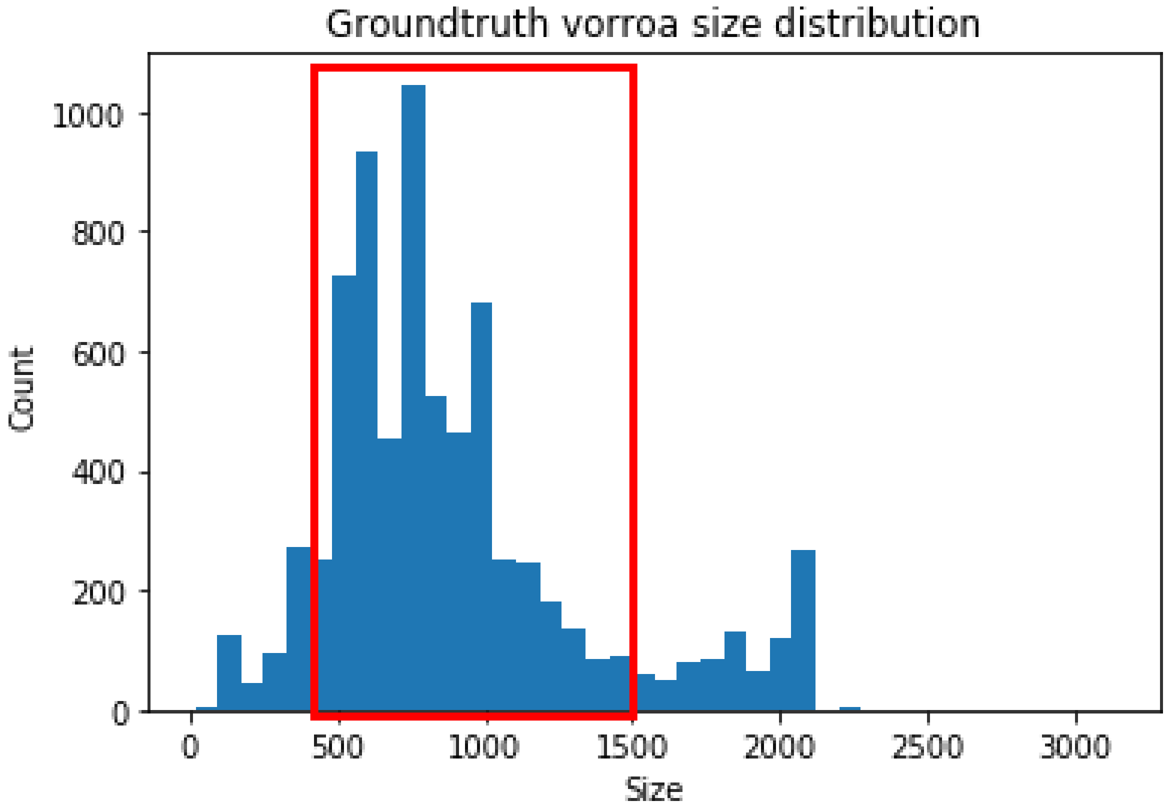


Segmentation with Edge detection


Edge detection segmentation

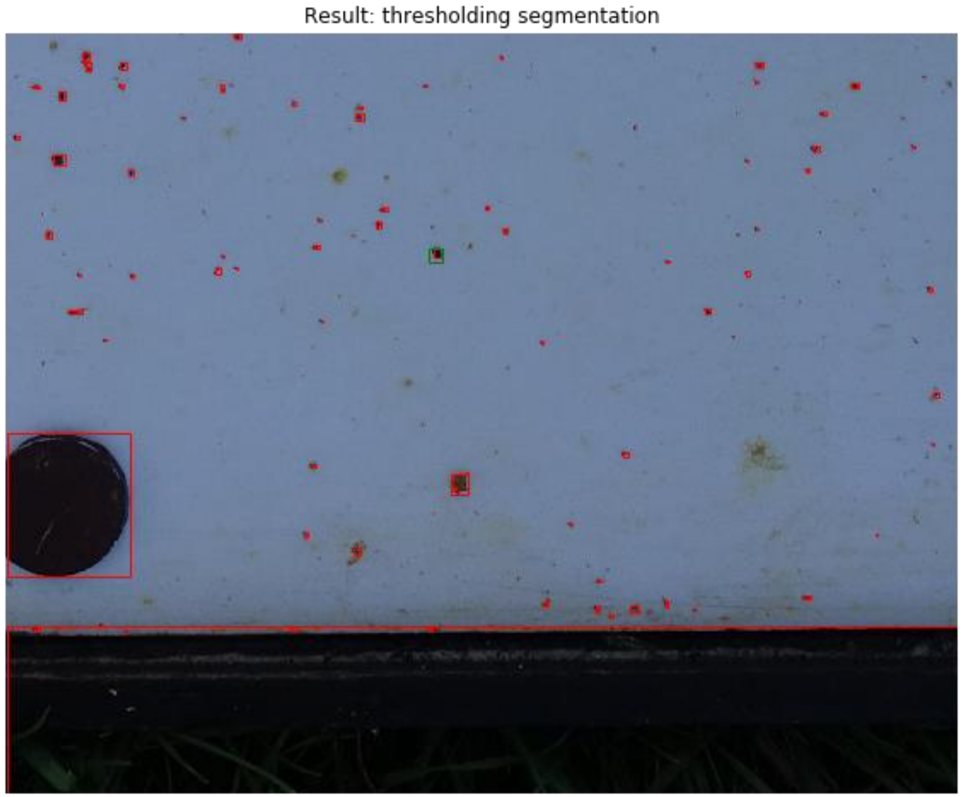

Result: segmentation with edge detection - Sobel filter

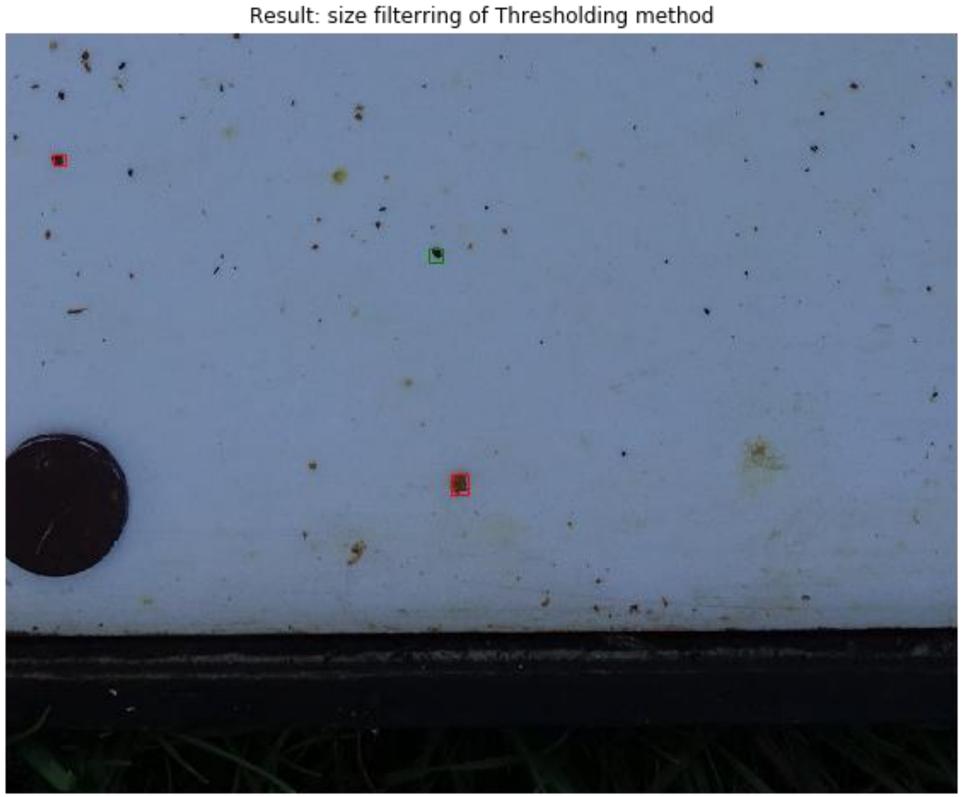
Ground truth varroa size distribution



Segmentation with Thresholding

With size filter

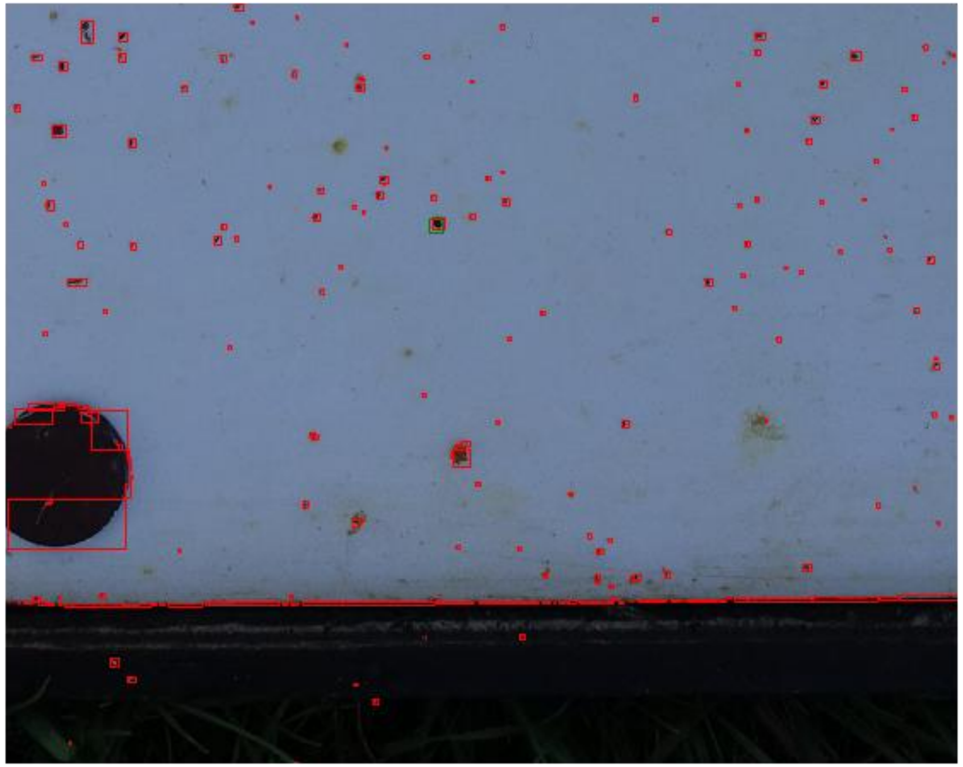




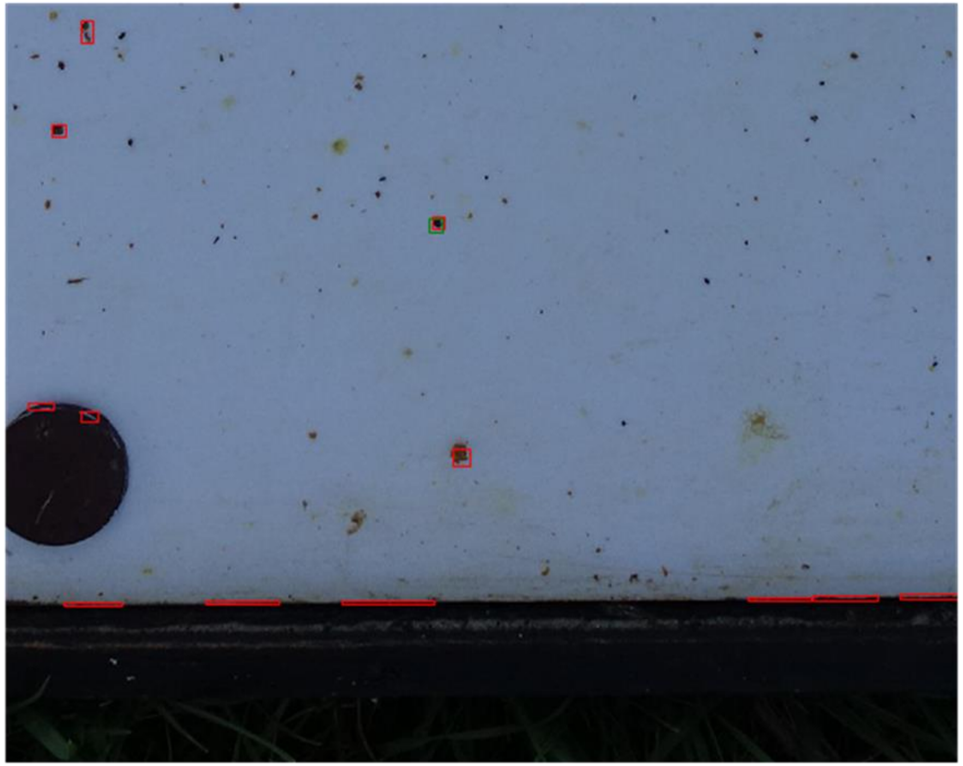
Segmentation with Edge detection

With size filter

Result: segmentation with edge detection - Sobel filter

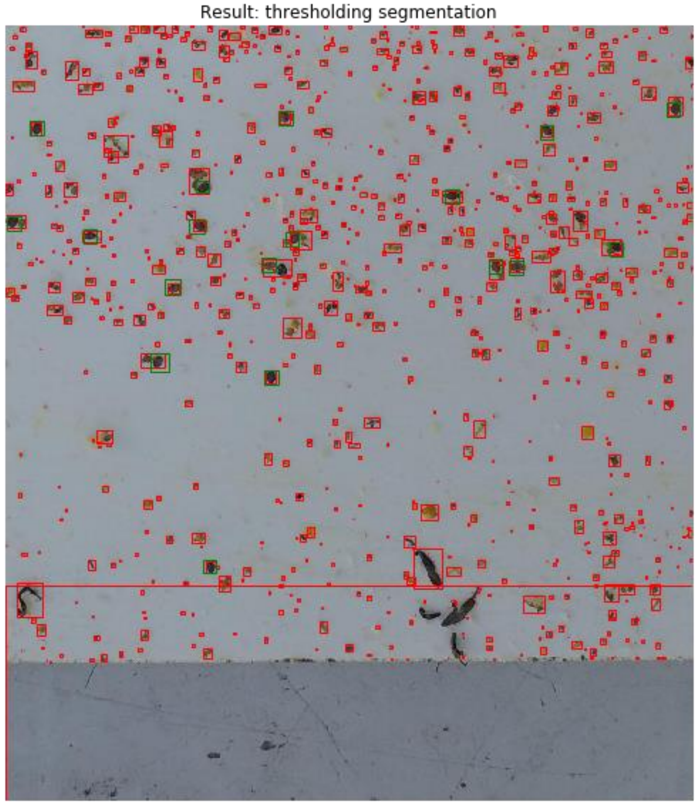


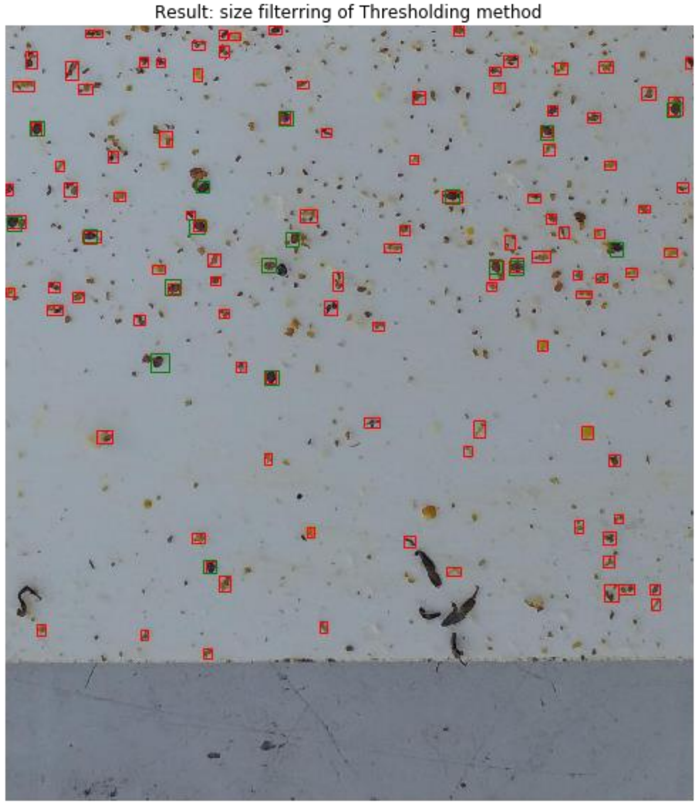
Result: size filtering of Edge detection method



Segmentation with Thresholding

With size filter

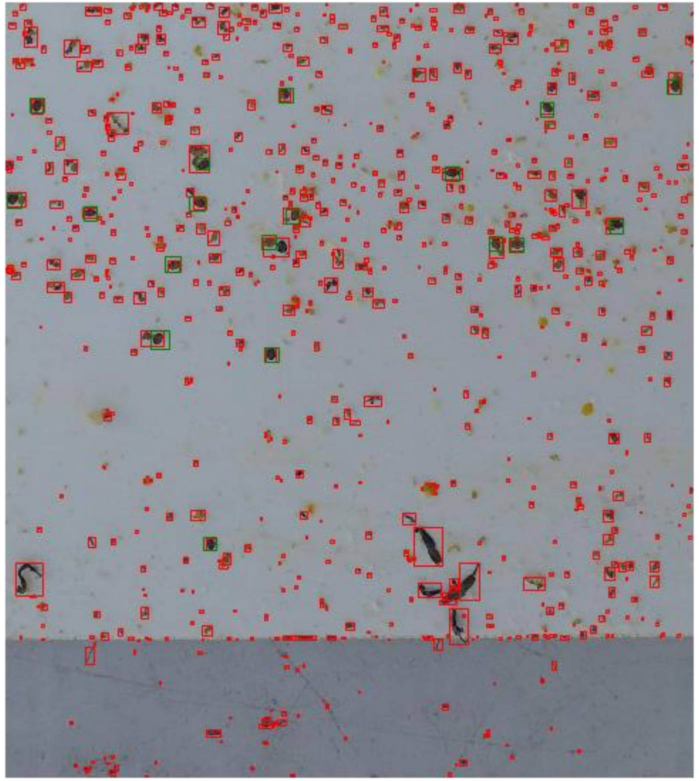




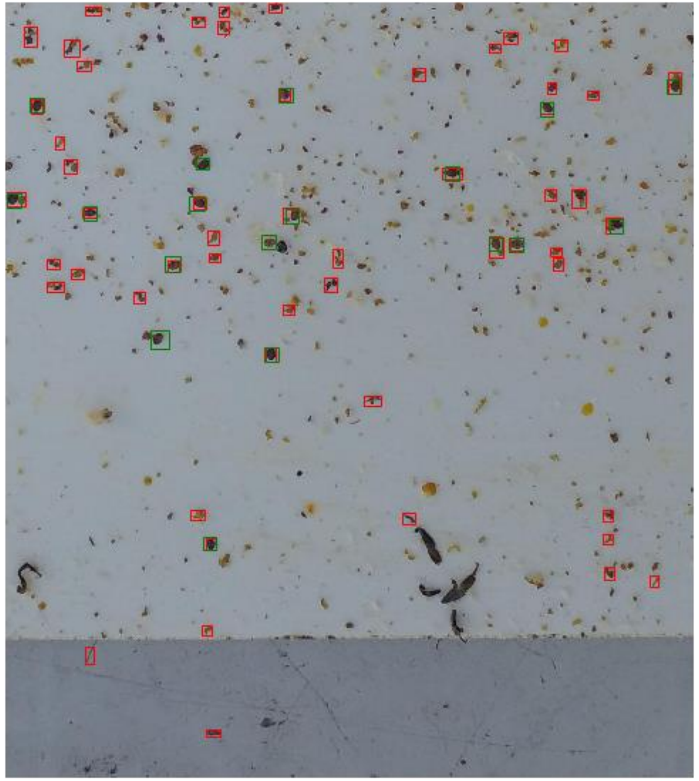
Segmentation with Edge detection

With size filter

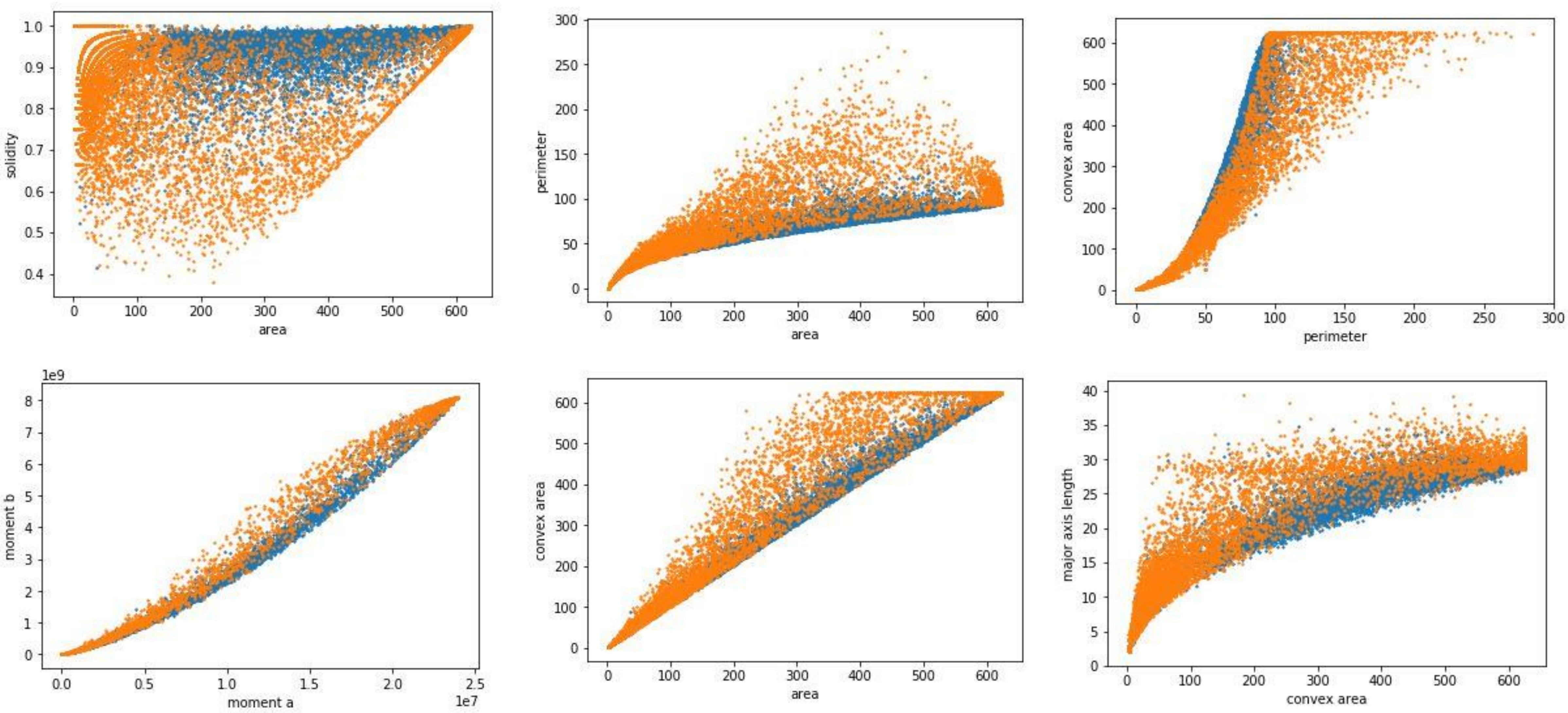
Result: segmentation with edge detection - Sobel filter



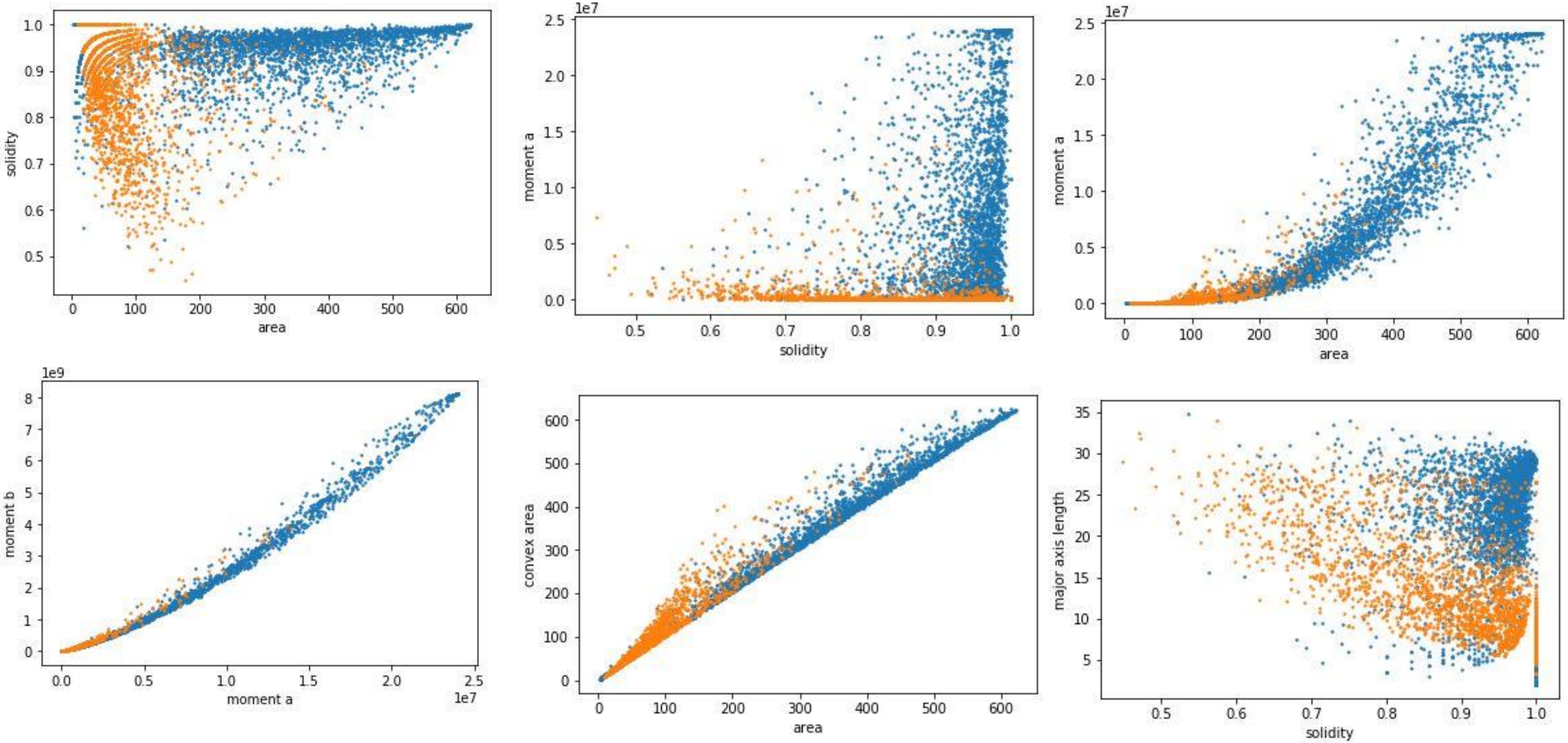
Result: size filtering of Edge detection method



Descriptors with thresholding method



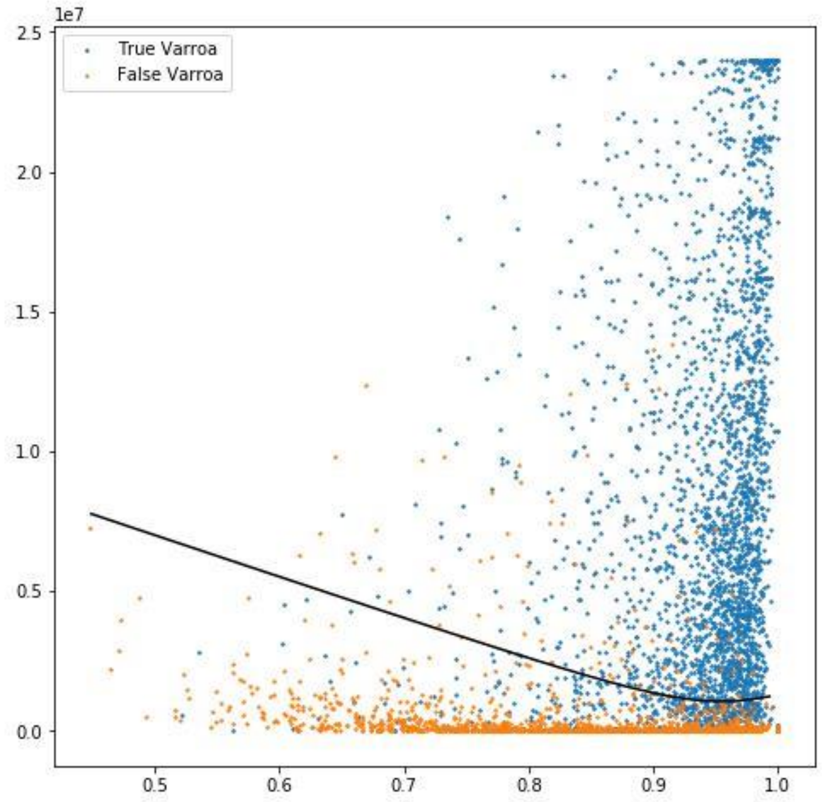
Descriptors with edge detection method



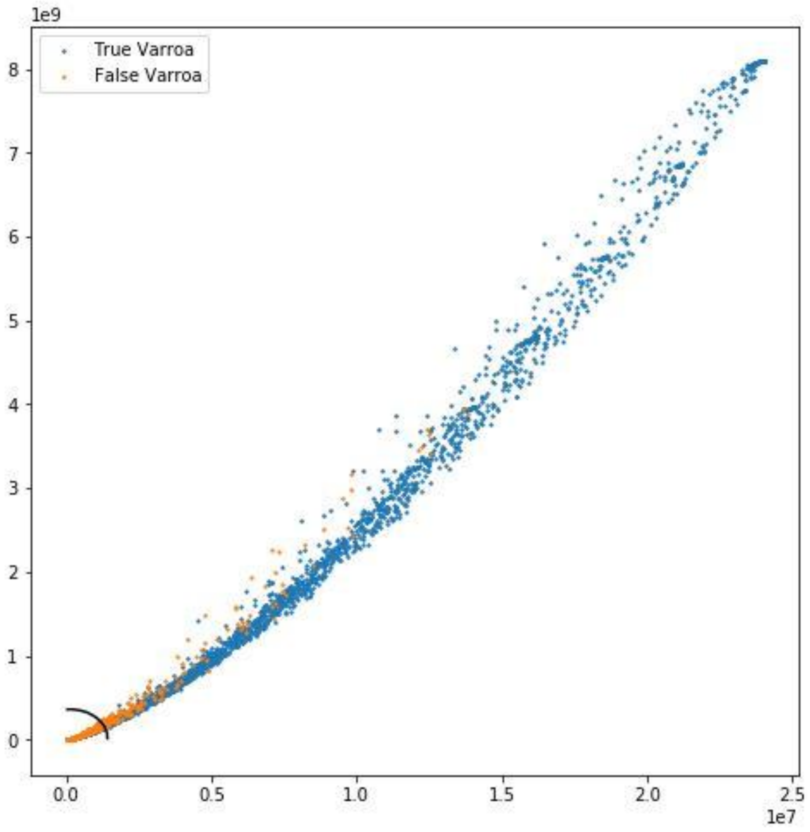
Bayes classifiers for edge segmentation

$$P(w_1 | x) \overset{?}{<>} P(w_2 | x)$$

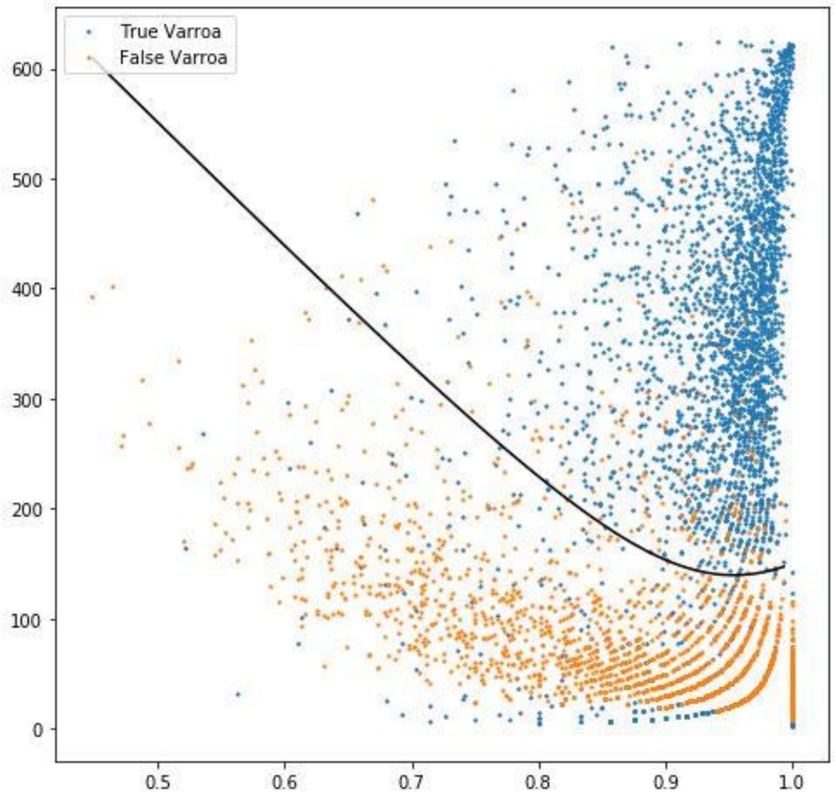
$$p(x | w_1)P(w_1) \overset{?}{<>} p(x | w_2)P(w_2)$$



Solidity vs convex area

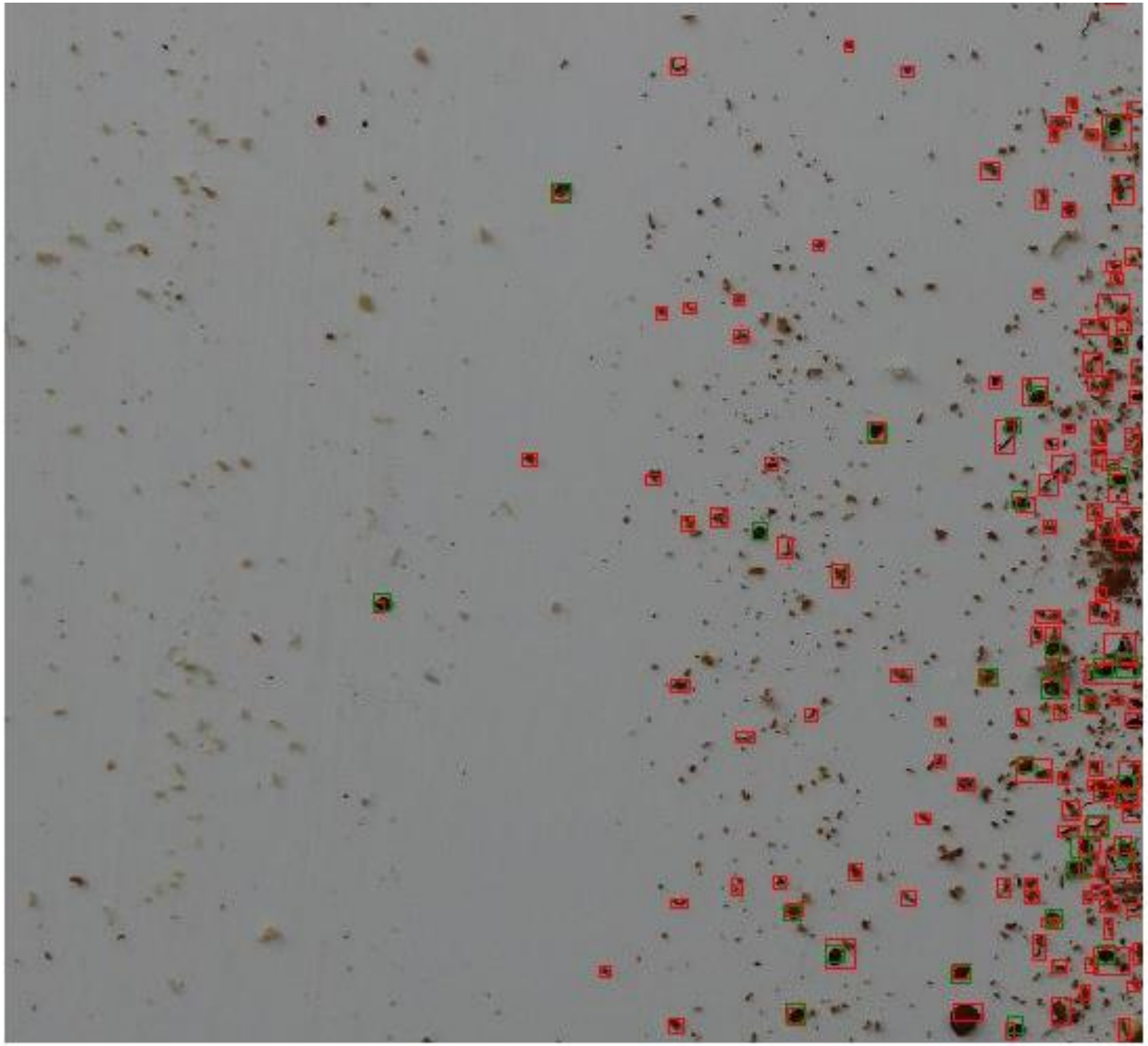


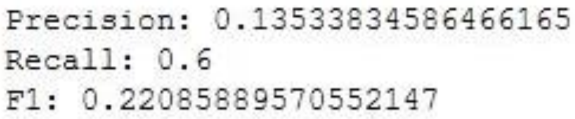
Moment 22 vs moment 33



Solidity vs moment 22

Results classification

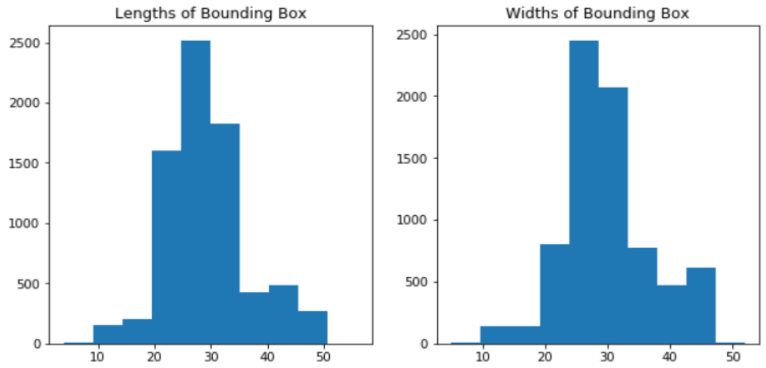




```
Precision: 0.13533834586466165
Recall: 0.6
F1: 0.22085889570552147
```

Sliding Window Classification

Create Dataset by Sliding Window



sliding window size: 25*25*3

1. **Sliding window without varroa:**

   Randomly crop 300 windows in each image;

   Not overlap with ground truth bounding box.

2. **Sliding window with varroa:**

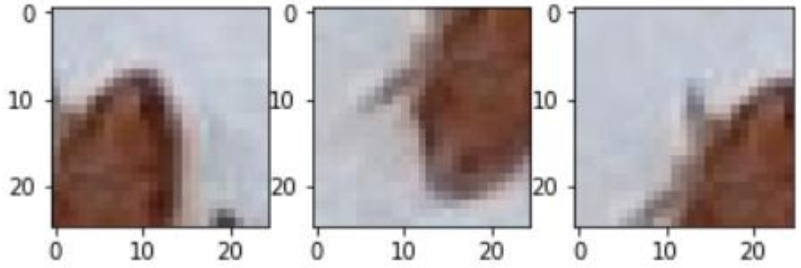
Do some transition to ground truth bbox:

① if bbox width >= size:

transition ∈ [-size/2, width-size/2]

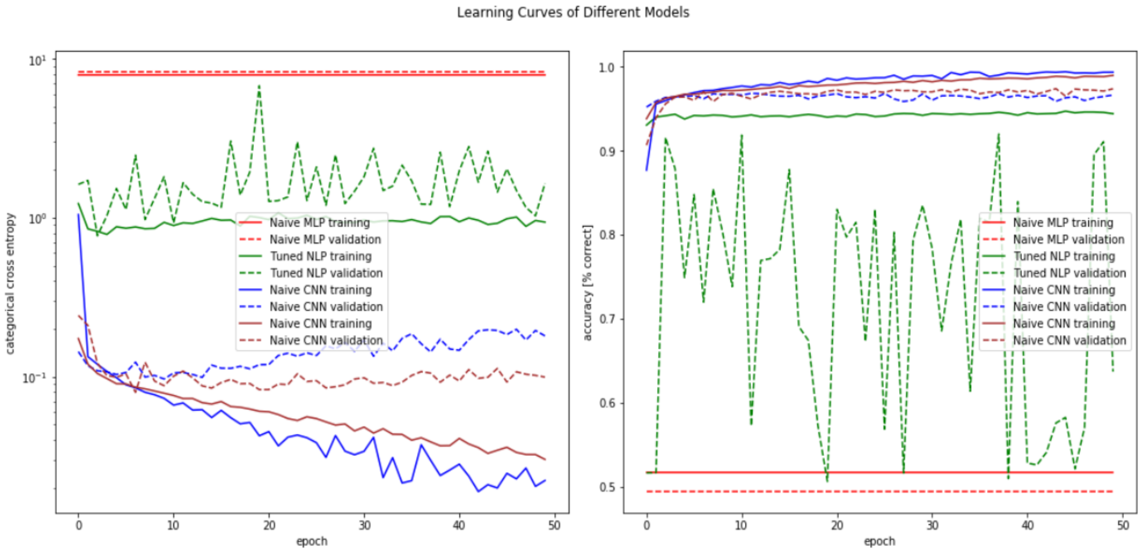
②if bbox width < size:

transition ∈ [width/2-size, width/2]



Sliding Window Classification

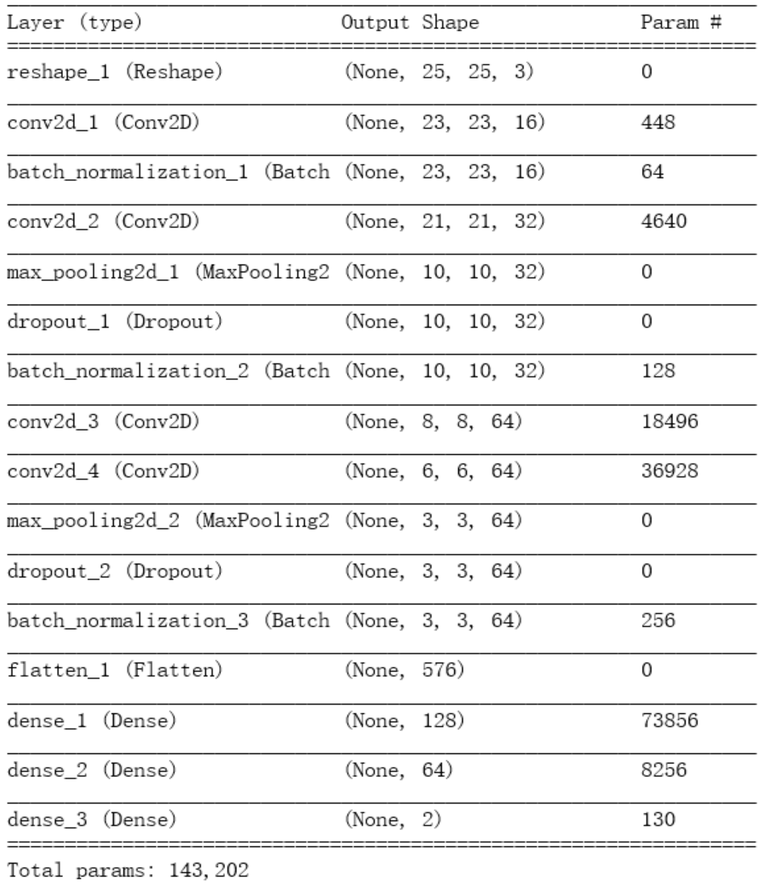
Choosing Best NN Model


Learning Curves of Different Models

Best Model: Tuned CNN

Highest accuracy and lowest cross entropy loss;

Less overfitting phenomenon.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
reshape_1 (Reshape)          (None, 25, 25, 3)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 23, 23, 16)        448
_________________________________________________________________
batch_normalization_1 (Batch (None, 23, 23, 16)        64
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 10, 32)        0
_________________________________________________________________
batch_normalization_2 (Batch (None, 10, 10, 32)        128
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 64)          18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 6, 6, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 3, 64)          0
_________________________________________________________________
batch_normalization_3 (Batch (None, 3, 3, 64)          256
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               73856
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 130
=================================================================
Total params: 143,202
```

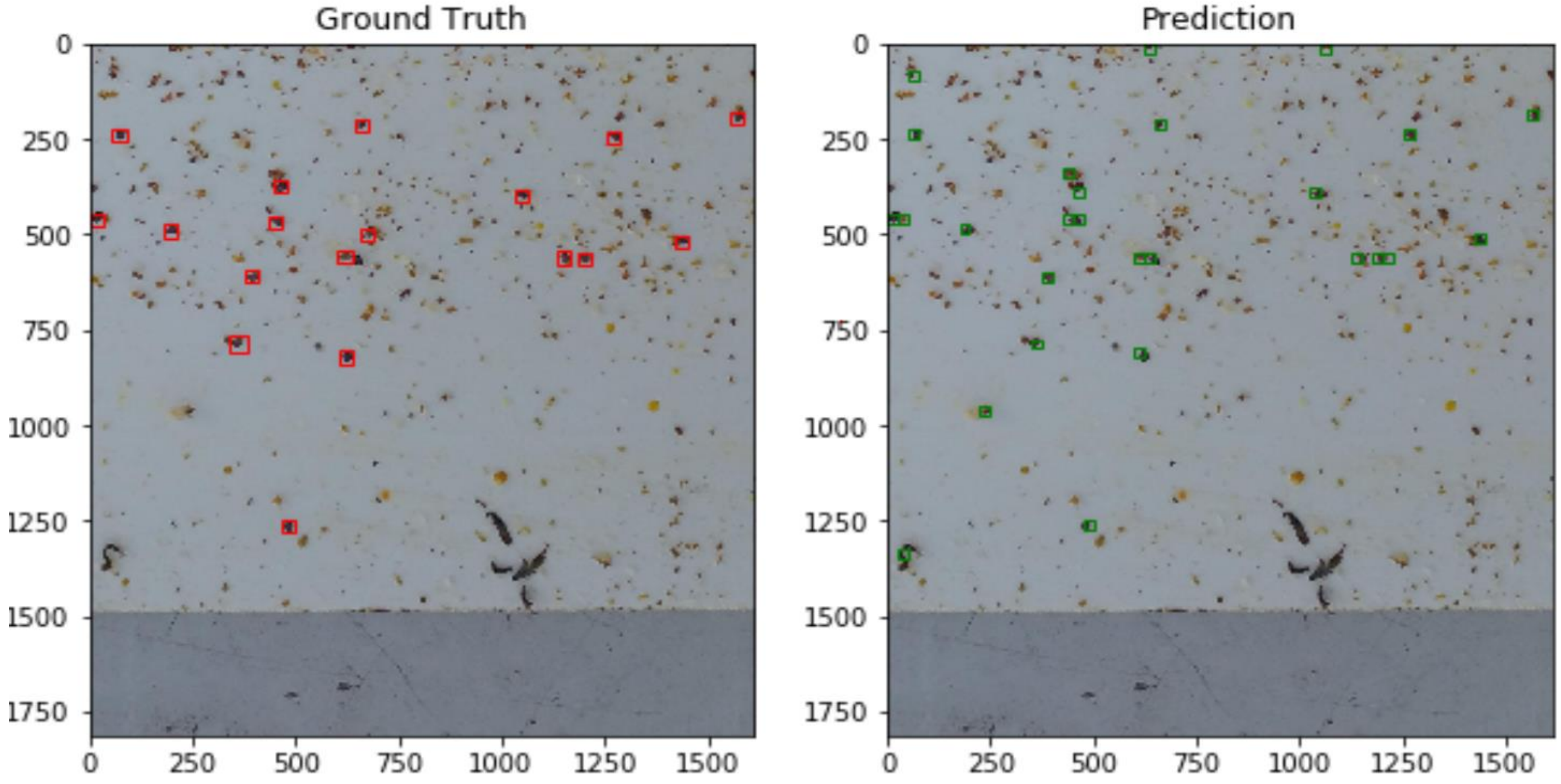
Sliding Window Classification

Prediction

- Train the CNN model on whole generated dataset
- Use transfer learning and early stopping to make training more efficient
- Iteratively crop each test image with 25*25*3 sliding window
- Predict the label of each potential bounding box
- Get the predicted bounding box with positive label

Sliding Window Classification

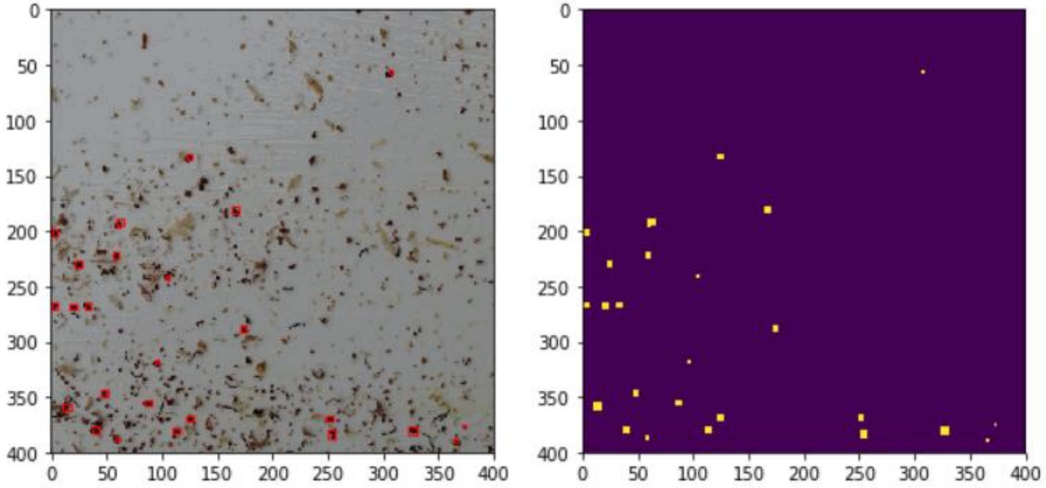
Prediction



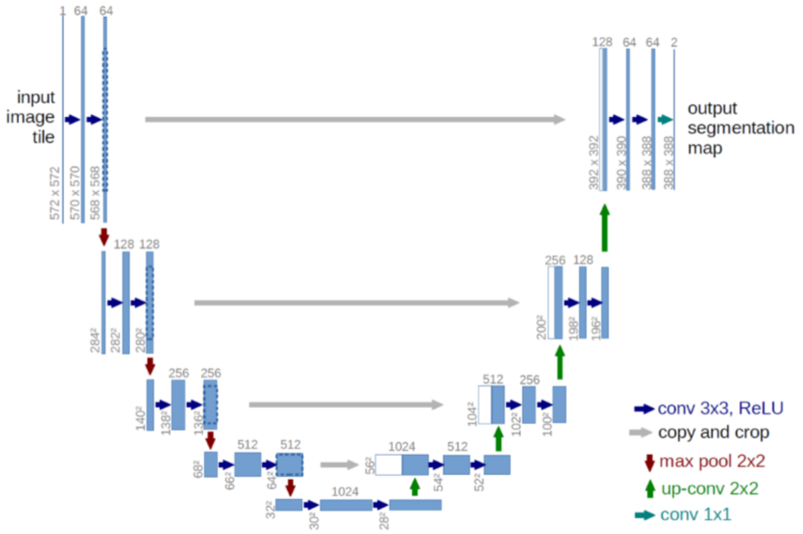
Binary Sermantic Segmentation

By U-Net

- **Create new dataset for sermantic segmentation:**
- **1. Resize all images to same scale**
- **2. Generate binary pixel label by bounding box**



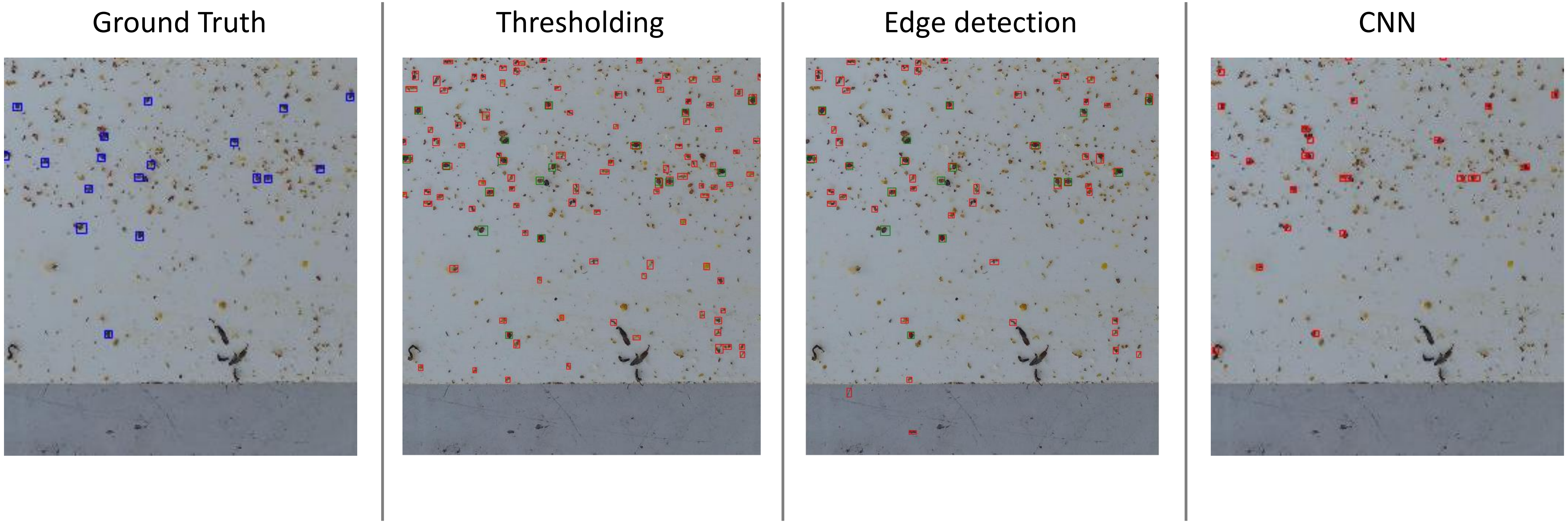
- **Build and train U-Net**



Dice Loss:

$$L_{Dice} = -\frac{\sum_i 2 o_i y_i}{\sum_i o_i + \sum_i y_i}$$

$$L_{Dice} \approx 1 - F_1$$



Ground Truth

Thresholding

Edge detection

CNN

Summary

- **Performance of different methods:**

	Precision	**Recall**	**F1 score**
Thresholding segmentation	0.24	0.61	0.35
Edge detection	0.15	0.68	0.24
Descriptor + Bayes method	0.10	0.26	0.15
Sliding window classification by CNN	0.30	0.60	0.38

Best F1 score on competition: 0.46

- **Future work:**
- **1. Enhance dataset**
- **2. Explore better segmentation and descriptors**
- **3. Try more advanced network in object detection:**
- **e.g. Yolo, Faster R-CNN, Open Pifpaf**
- **4. Implement more precise hyperparameter tuning**
- **5. Make good use of priori knowledge**
- **i.e. combine machine learning methods with proper pre-processing**