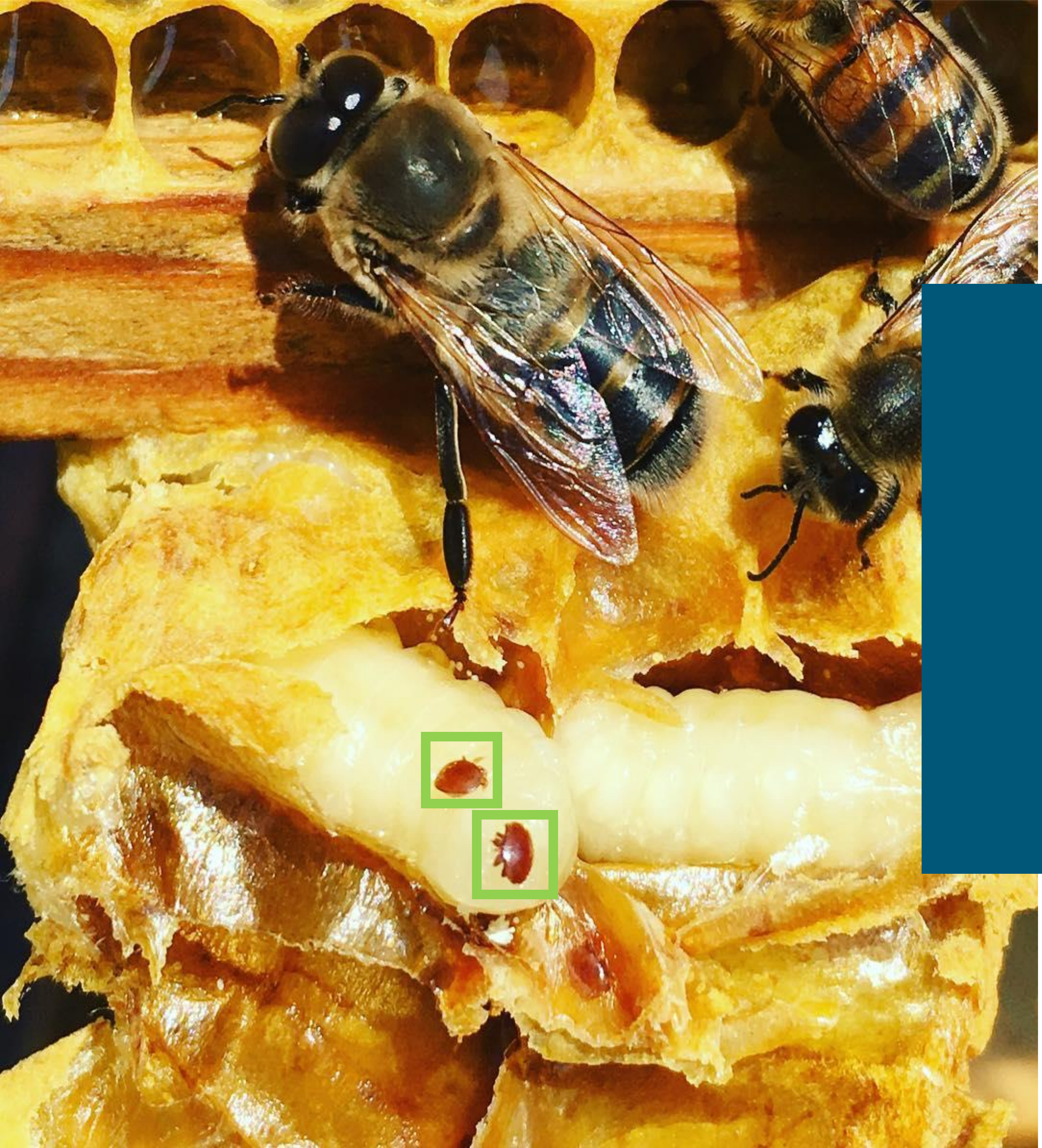
Varroa Detection in Beehives

Fengyu Cai, Wanhao Zhuo, Yamin Sepehri

Under Supervision of Prof. J.P. Thiran

Image Analysis and Pattern Recognition

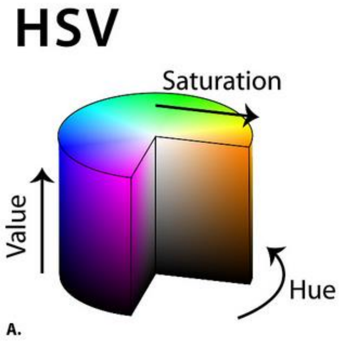
2019



Goal

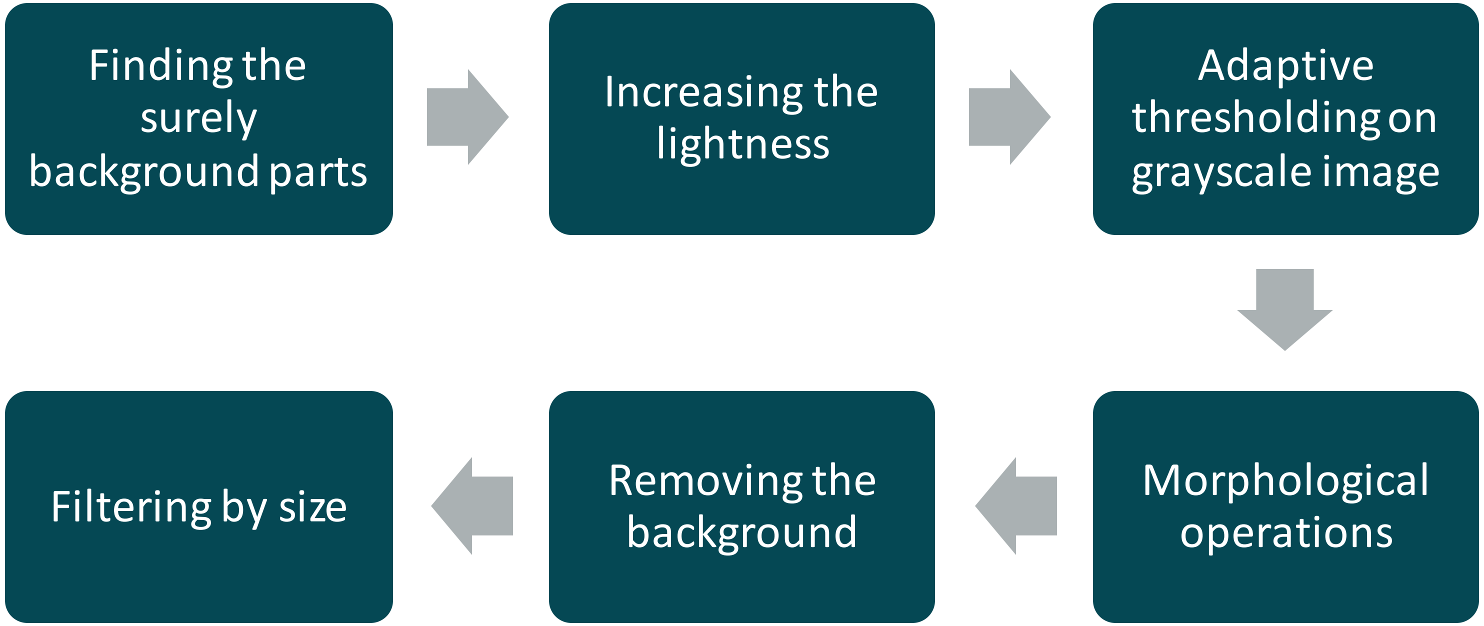
- Studying the different methods of detection in a real problem.
- Comparing classical image segmentation, sliding window-based classification, and convolutional neural network.

Image segmentation



Adaptive Method:

_ADAPTIVE_THRESH_MEAN_C



Finding the surely background parts

Increasing the lightness

Adaptive thresholding on grayscale image

Morphological operations

Removing the background

Filtering by size

- Opening
- Closing
- Remove Small Objects
- Fill Holes

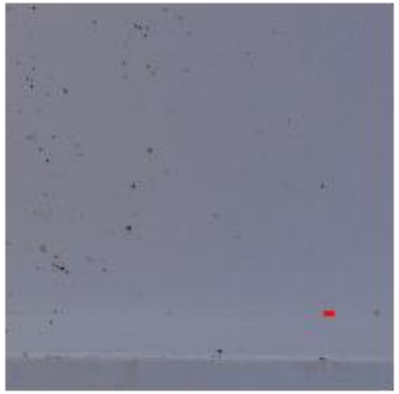
Image segmentation

Hyperparameter Tuning

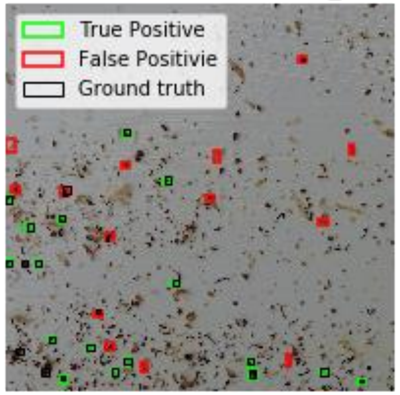
Based on Train Set

- Lightness improvement
- Morphological operations
- Size Filter Criteria
- ...

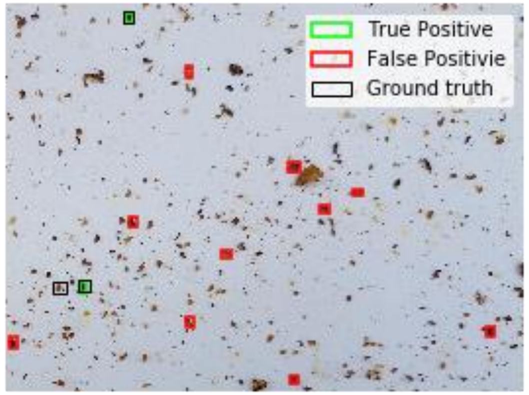
Precision= 0.0 , Recall=XX, F1_score=XX



Precision= 0.548 , Recall=0.654, F1_score=0.596



Precision= 0.167 , Recall=0.667, F1_score=0.267



Precision= 0.0 , Recall=XX, F1_score=XX

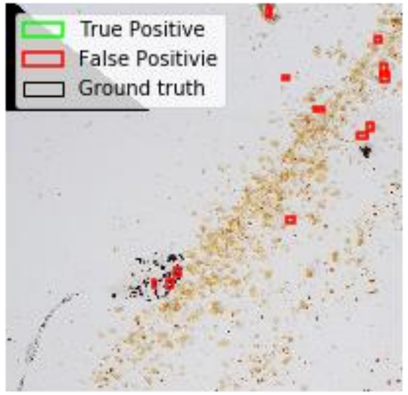
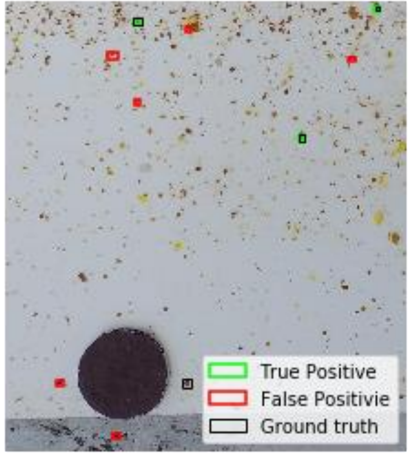


Image segmentation

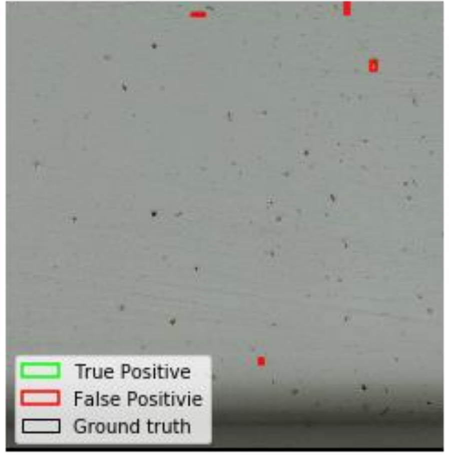
Final

Testing

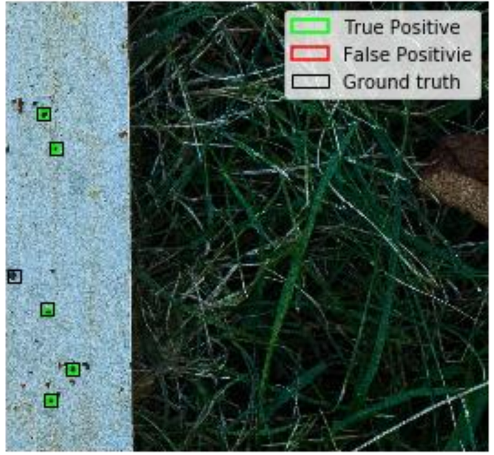
Precision= 0.333 , Recall=0.75, F1_score=0.462



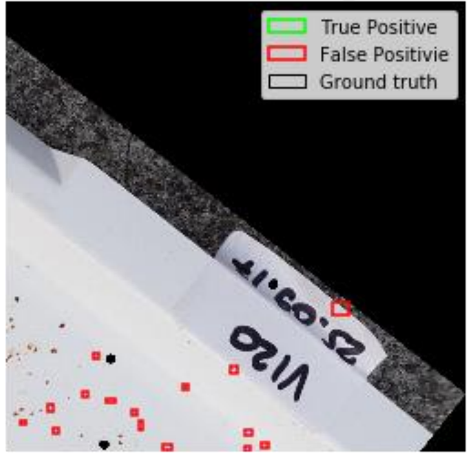
Precision= 0.0 , Recall=XX, F1_score=XX

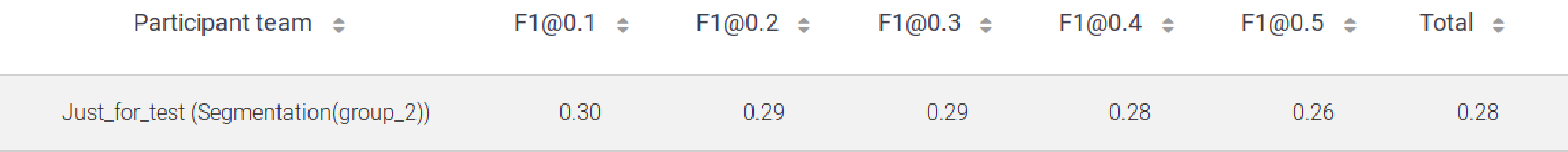


Precision= 1.0 , Recall=0.833, F1_score=0.909



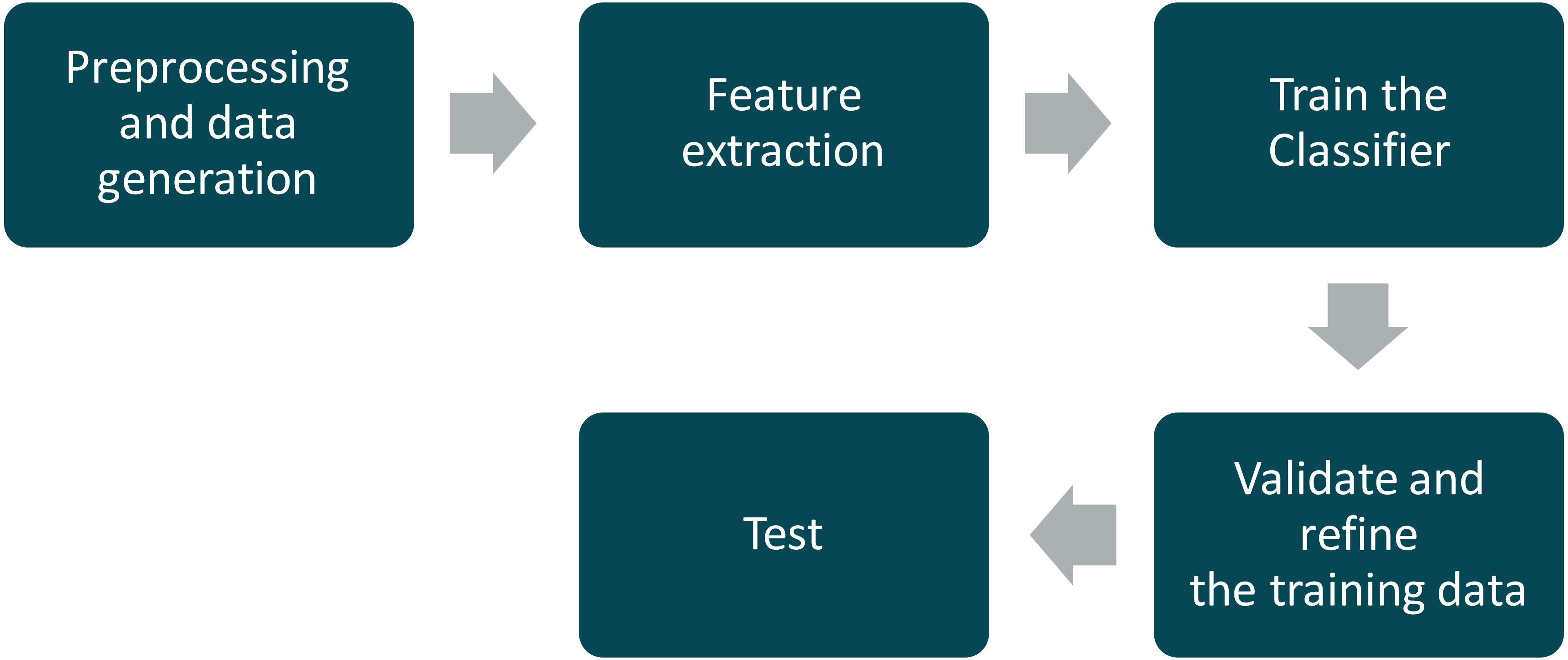
Precision= 0.0 , Recall=XX, F1_score=XX





Participant team	F1@0.1	F1@0.2	F1@0.3	F1@0.4	F1@0.5	Total
Just_for_test (Segmentation(group_2))	0.30	0.29	0.29	0.28	0.26	0.28

Sliding-window based Classifier: Methodology



Preprocessing and data generation

Feature extraction

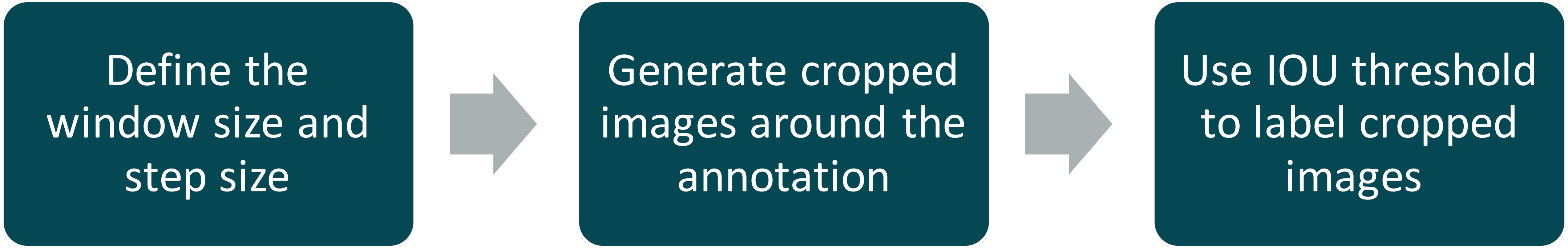
Train the Classifier

Validate and refine the training data

Test

Sliding-window based Classifier: Methodology

- **Step 1: Preprocessing and generate training data**



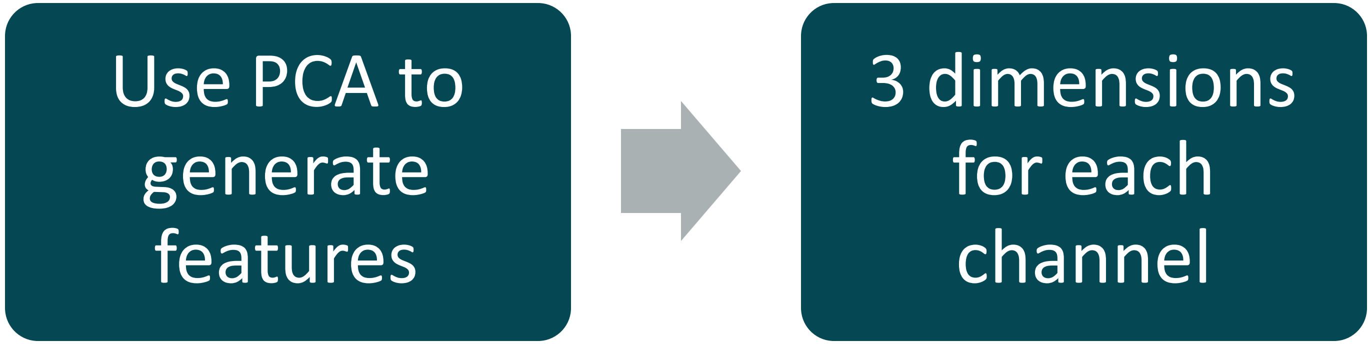
Define the window size and step size

Generate cropped images around the annotation

Use IOU threshold to label cropped images

Sliding-window based Classifier: Methodology

- **Step 3: Feature extraction**

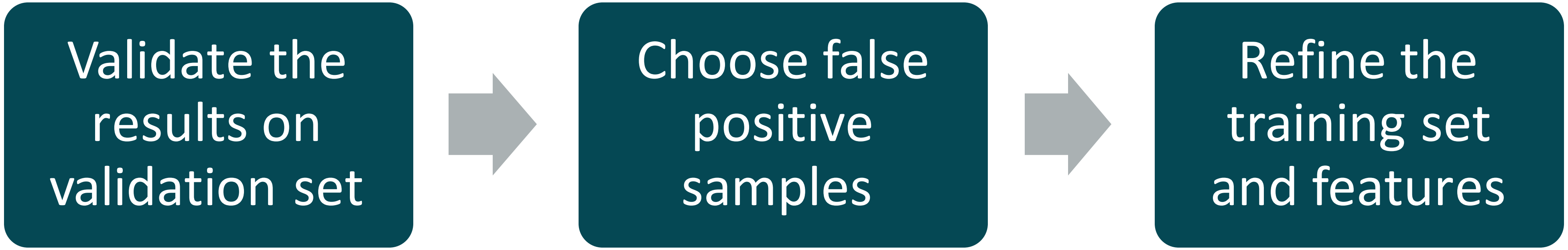


Use PCA to generate features

3 dimensions for each channel

Sliding-window based Classifier: Methodology

- **Step 4: Train Classifier (Naïve Bayes, SVM, GBDT)**
- **Step 5: Validate and refine the training data**



Validate the results on validation set

Choose false positive samples

Refine the training set and features

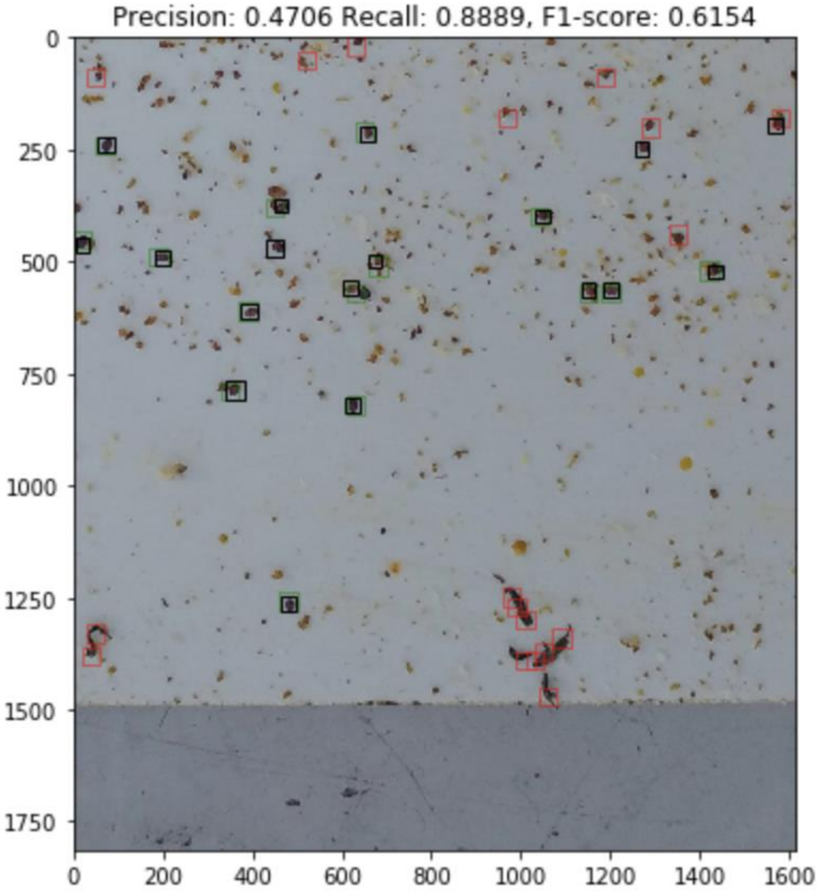
Sliding-window based Classifier: Methodology

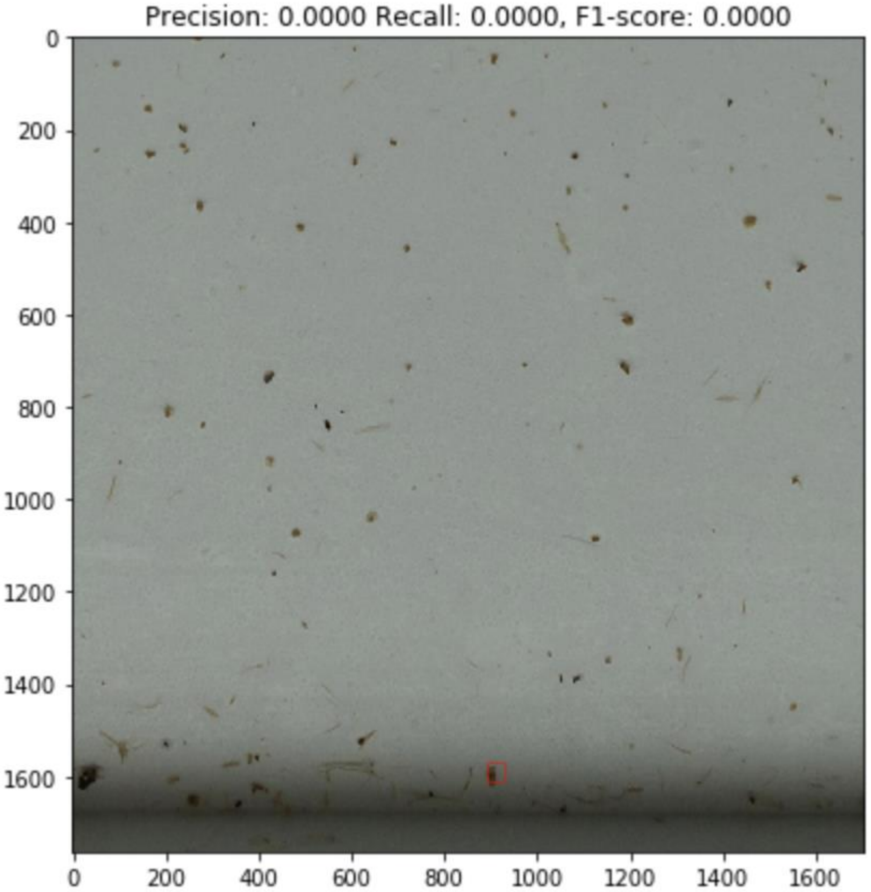
- Step 6: Test
  - Run the test and observe the detected windows
  - Global F1 score at around 0.23
  - Global Precision is still relatively low ~0.14

Sliding-window based Classifier: Results and analysis

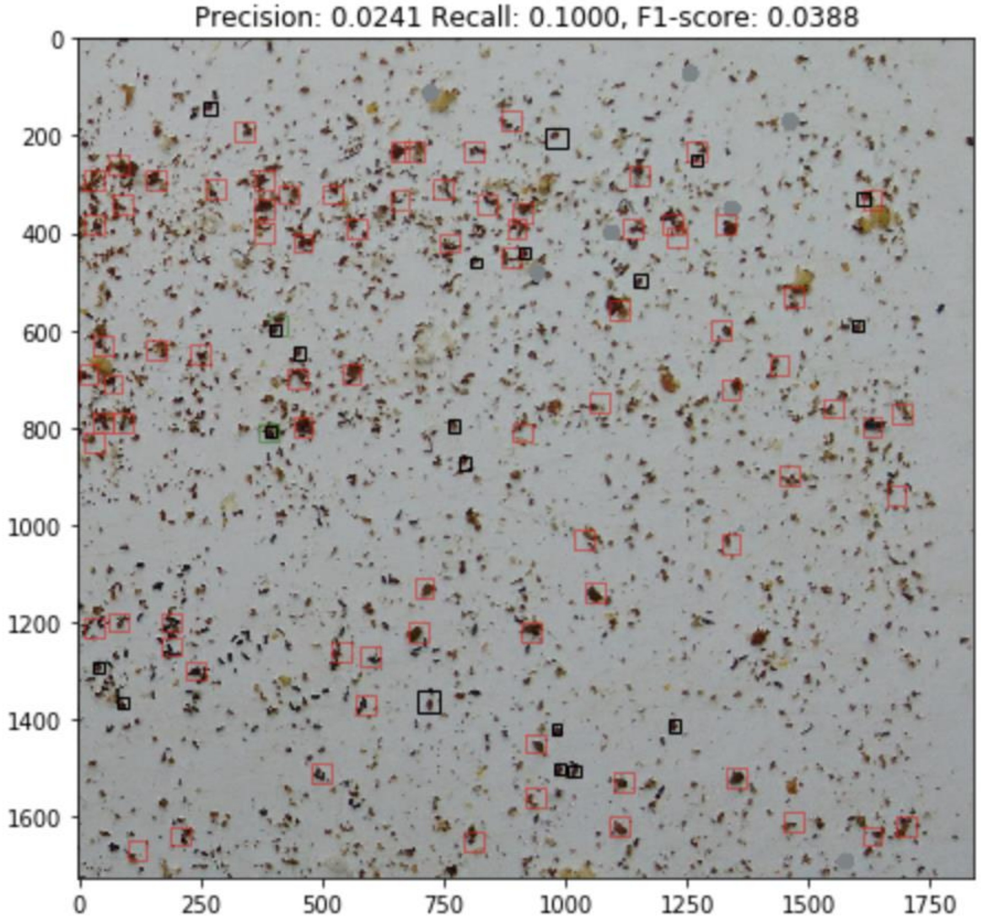
- Pros
  - Easy to train
  - Intuitive
  - Clear results on "clean" data
- Cons

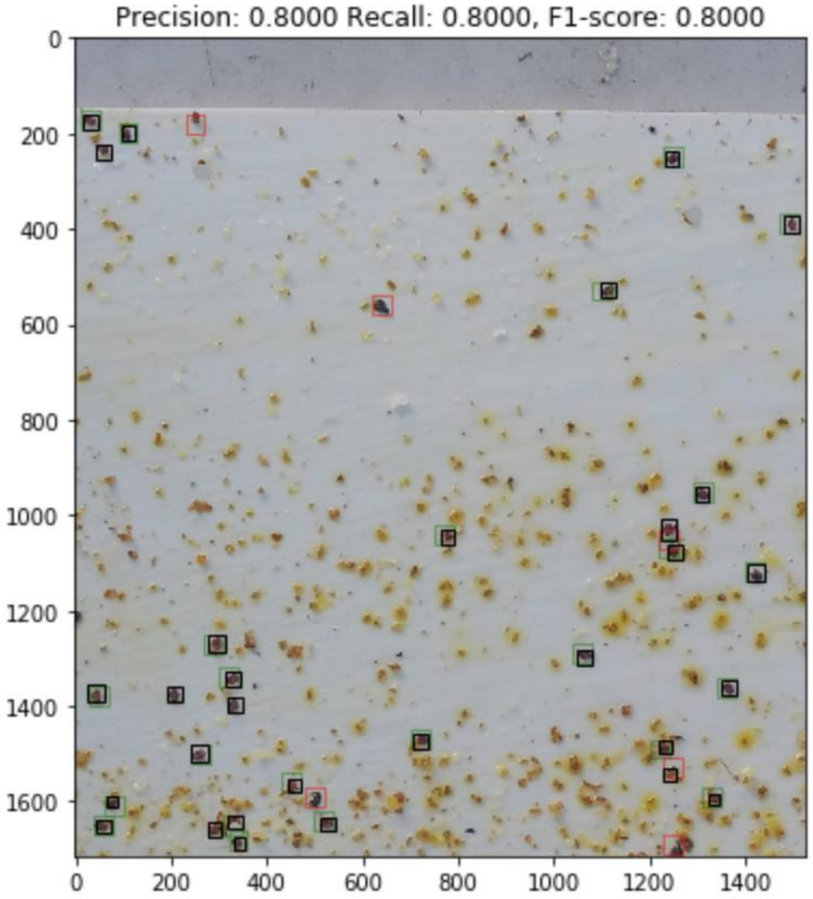
Sliding-window based Classifier: Results and analysis





Sliding-window based Classifier: Results and analysis


Precision: 0.0241 Recall: 0.1000, F1-score: 0.0388


Precision: 0.8000 Recall: 0.8000, F1-score: 0.8000

MLP and CNN Method

- Sliding-window based methods
  - Simple CNN
  - VGG-16

- Region propose methods
  - Mask R-CNN

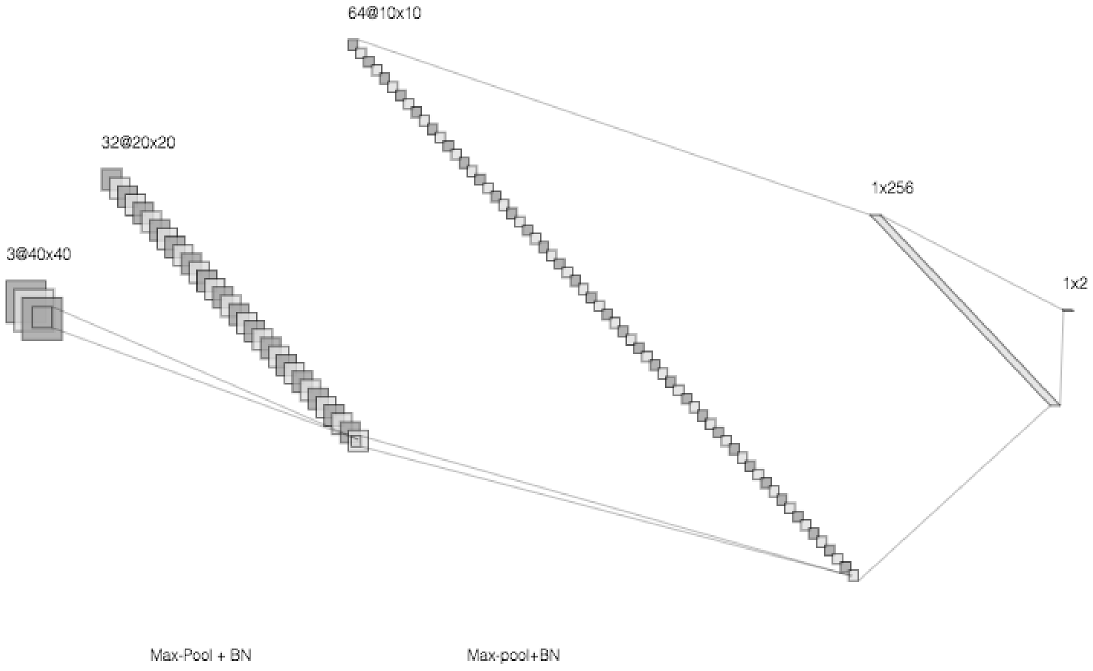
MLP and CNN Method – Sliding-window based methods

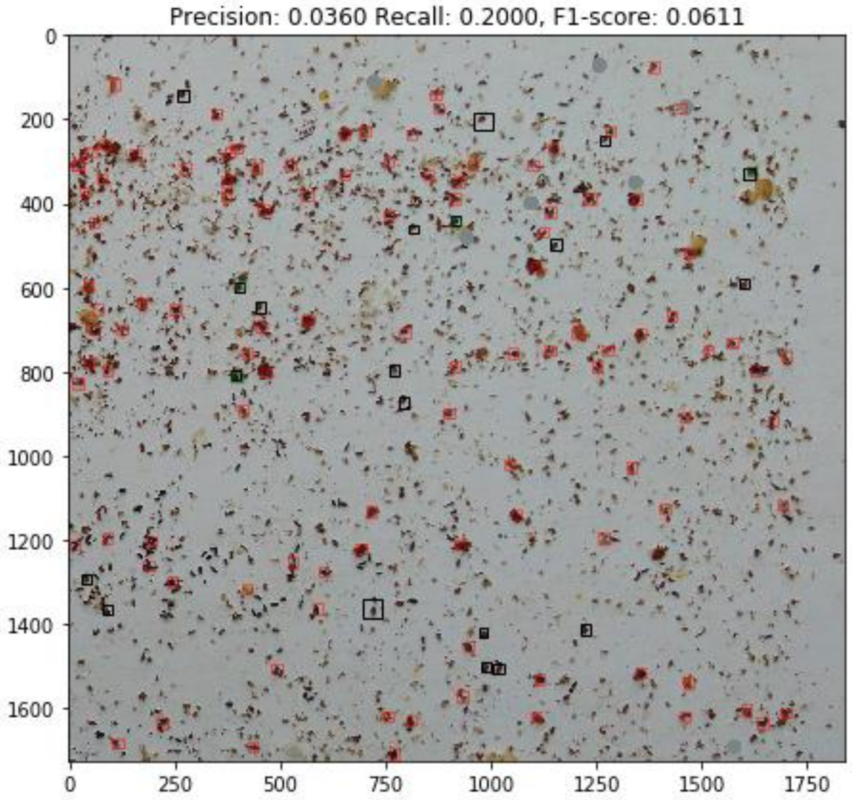
Similar with the sliding-window based classifier：

- Slide on the image to generate positive and negative windows feed in the neural network framework
- Start from the basic 2-layer CNN
- Follow with fully connected layers and end with 2 unit
- Generate probability with Softmax

MLP and CNN Method – Sliding-window based methods

Architecture of 2-layer CNN and some results





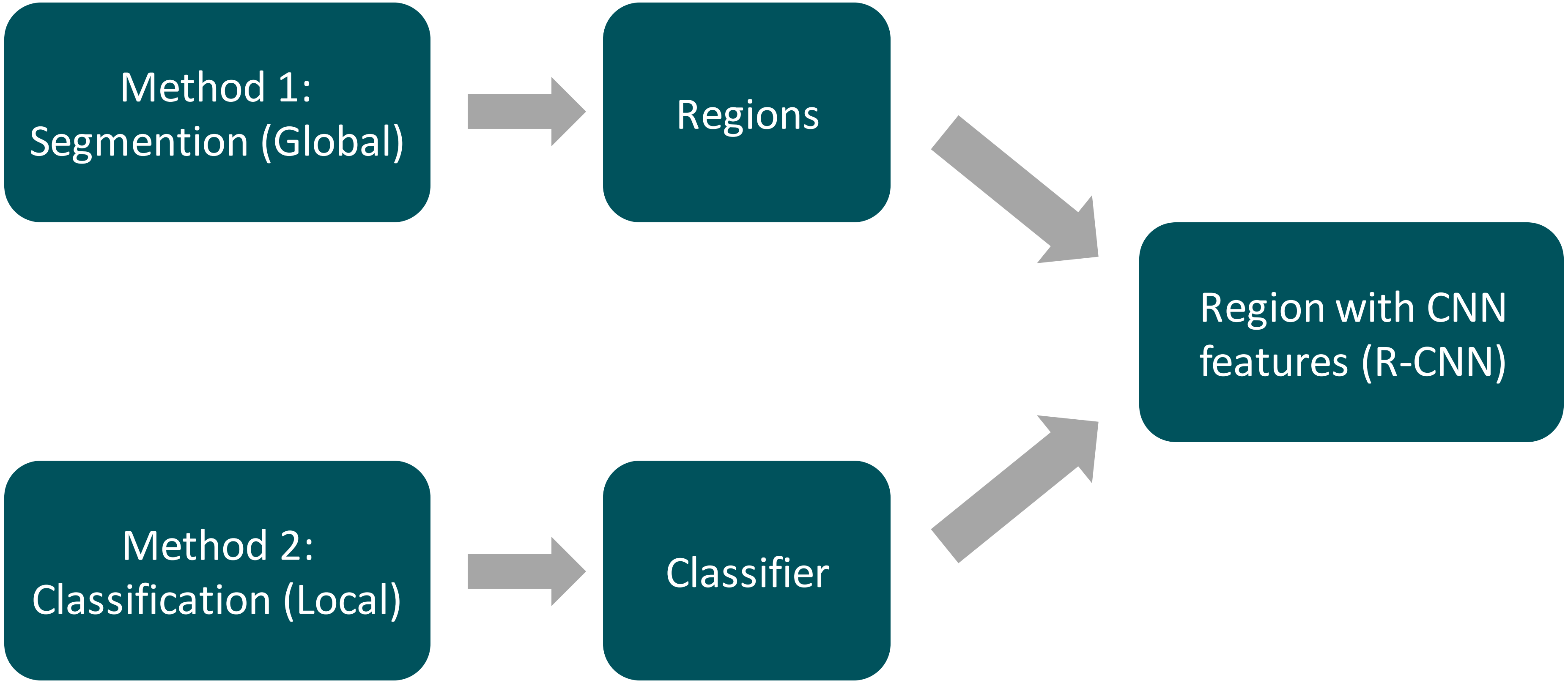
MLP and CNN Method – Reflection about previous methods

- Potential reason I: the network is not complicated enough!
- Solution: VGG-16
- Result: Still poor, and training is slow

MLP and CNN Method – Reflection about previous methods

- Potential reason II: window size is fixed
- Solution: multiple choices of window size: 24px, 36px and 48px
- Another problem: block overlapping
- Solution: Non-maximum suppression
- Result: Still poor, no improvement from method 2

MLP and CNN Method – Region propose method



Method 1:
Segmention (Global)

Regions

Method 2:
Classification (Local)

Classifier

Region with CNN
features (R-CNN)

MLP and CNN Method – Procedure of R-CNN

I. Extract Regions of Interest (RoI): selective search

II. CNN feature extraction on each RoI

III. SVM classifier

IV. Bounding Box Regression

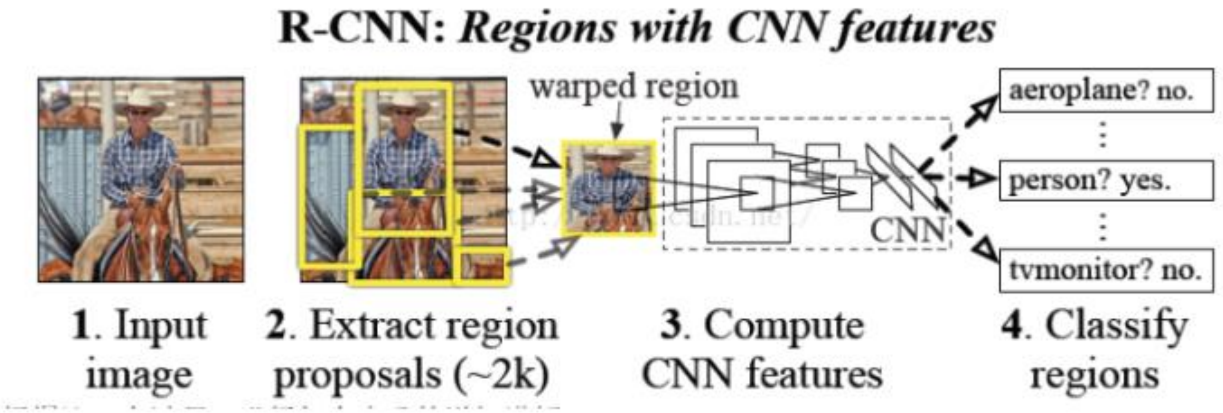
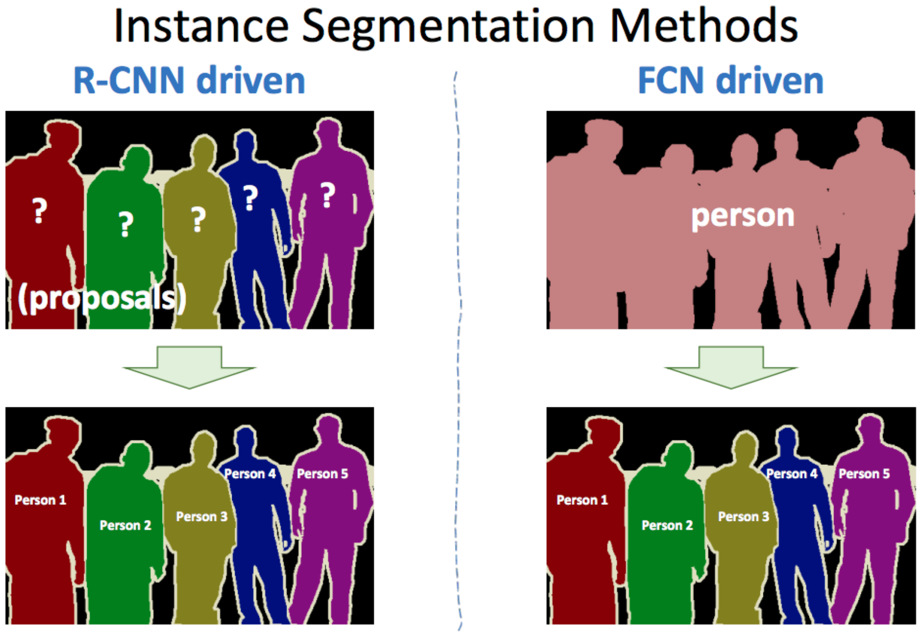


Image Source: Rich feature hierarchies for accurate object detection and semantic segmentation, R.Girshick J.Donahue T. Darrell J,Malik, UC Berkeley

MLP and CNN Method – Derivatives of R-CNN

- Fast R-CNN: Extract the feature of RoI from the feature map of the input image
- Faster R-CNN: Region Proposal Network (RPN) substitutes the previous selective search
- Mask R-CNN: the method we used

MLP and CNN Method – Mask R-CNN



Mask R-CNN

- Mask R-CNN = **Faster R-CNN** with **FCN** on RoIs

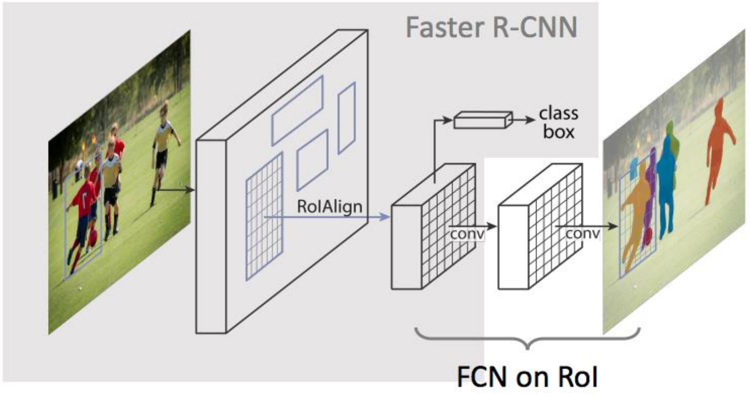
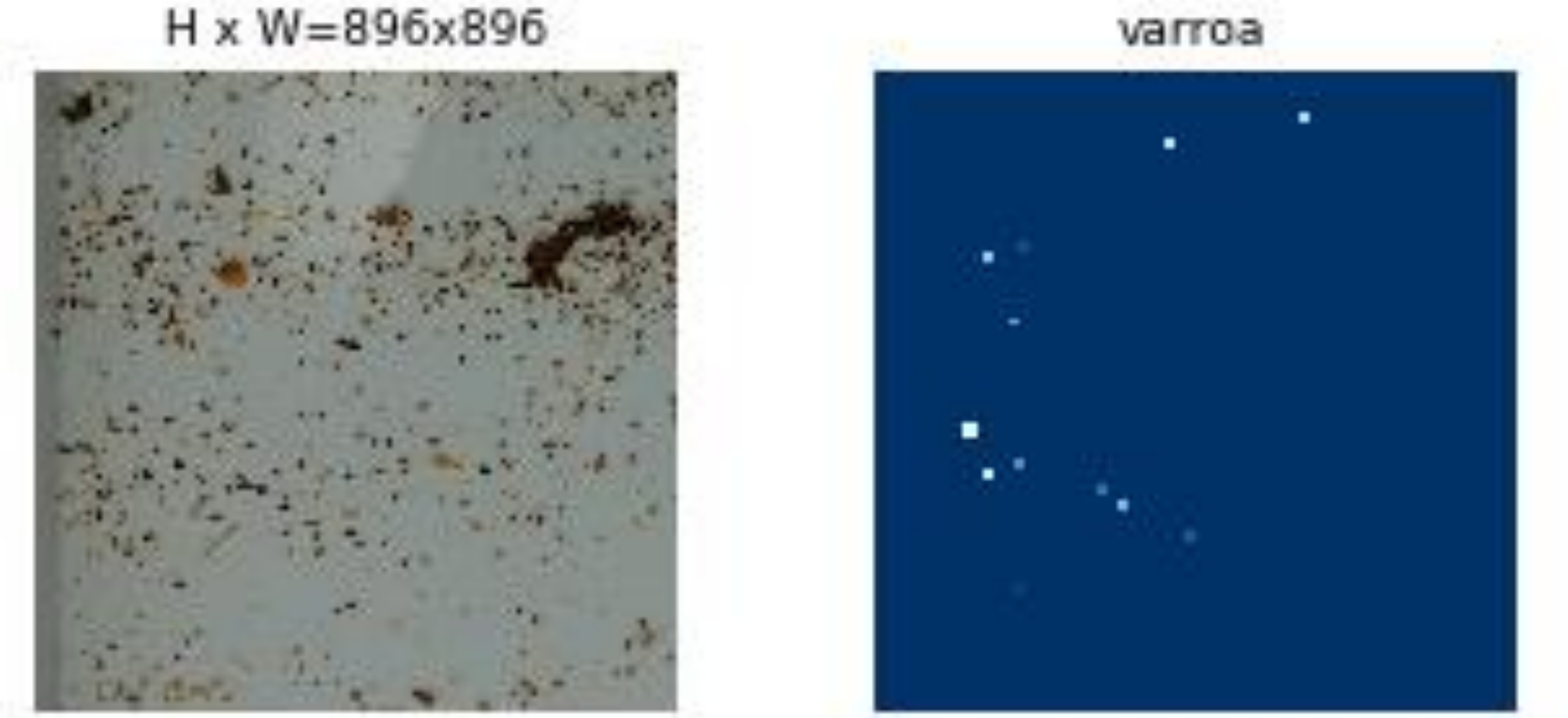


Image Source: http://kaiminghe.com/iccv17tutorial/maskrcnn_iccv2017_tutorial_kaiminghe.pdf

MLP and CNN Method – Mask R-CNN: Generate the masks

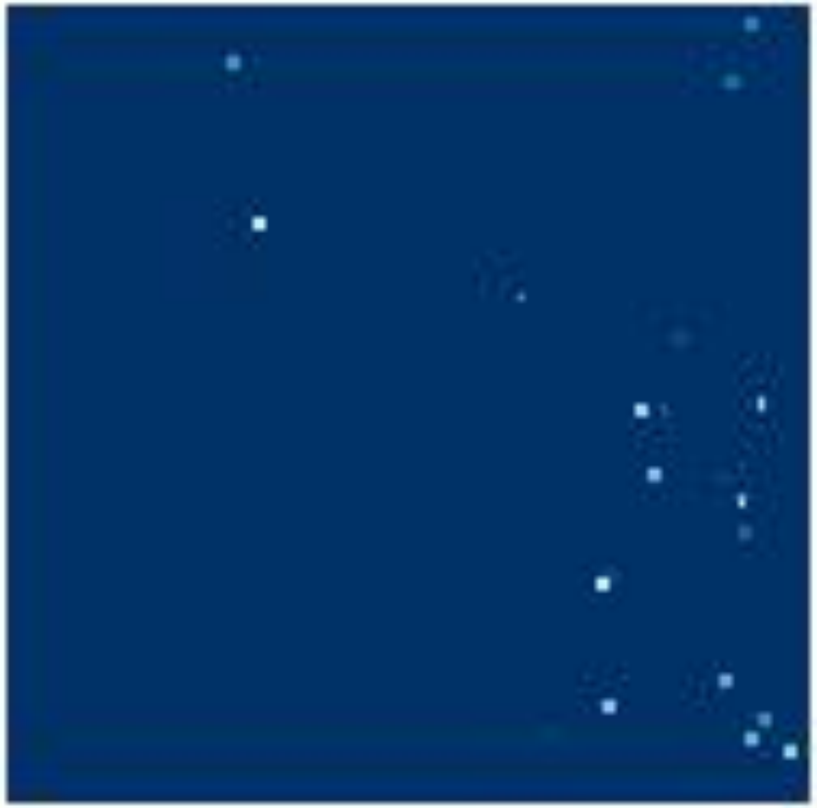


H x W=896x896

varroa

MLP and CNN Method – Mask R-CNN: Generate the masks

H x W=896x896



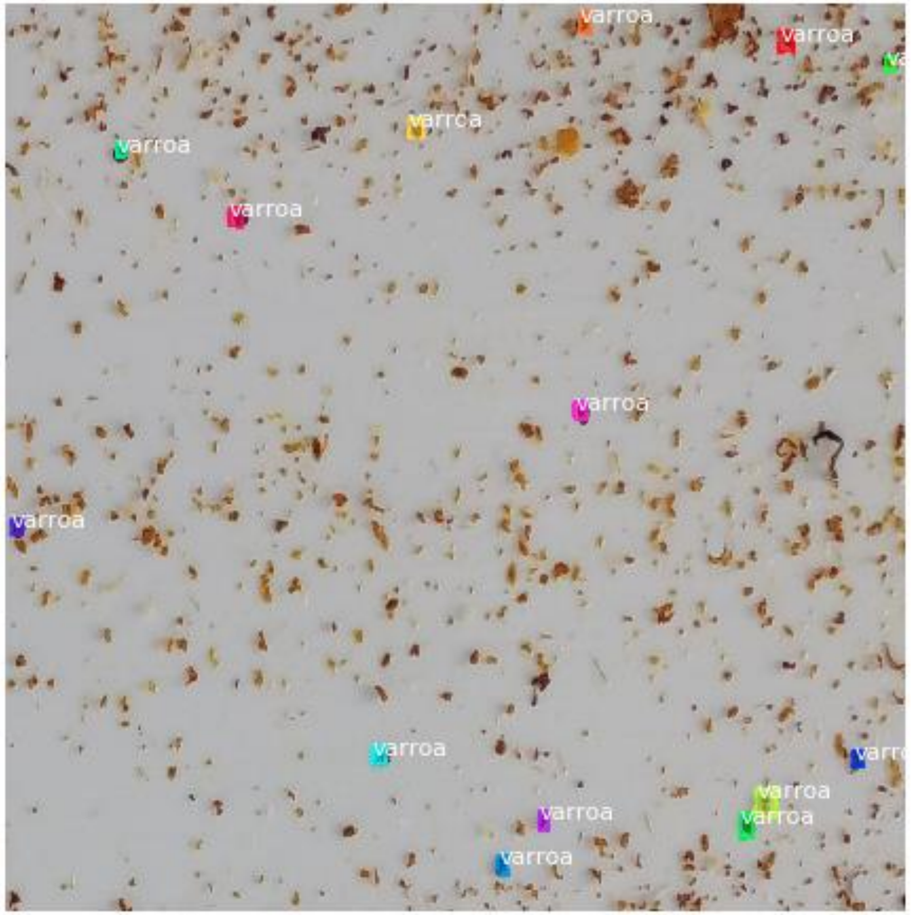
varroa

MLP and CNN Method – Mask R-CNN: Train the model

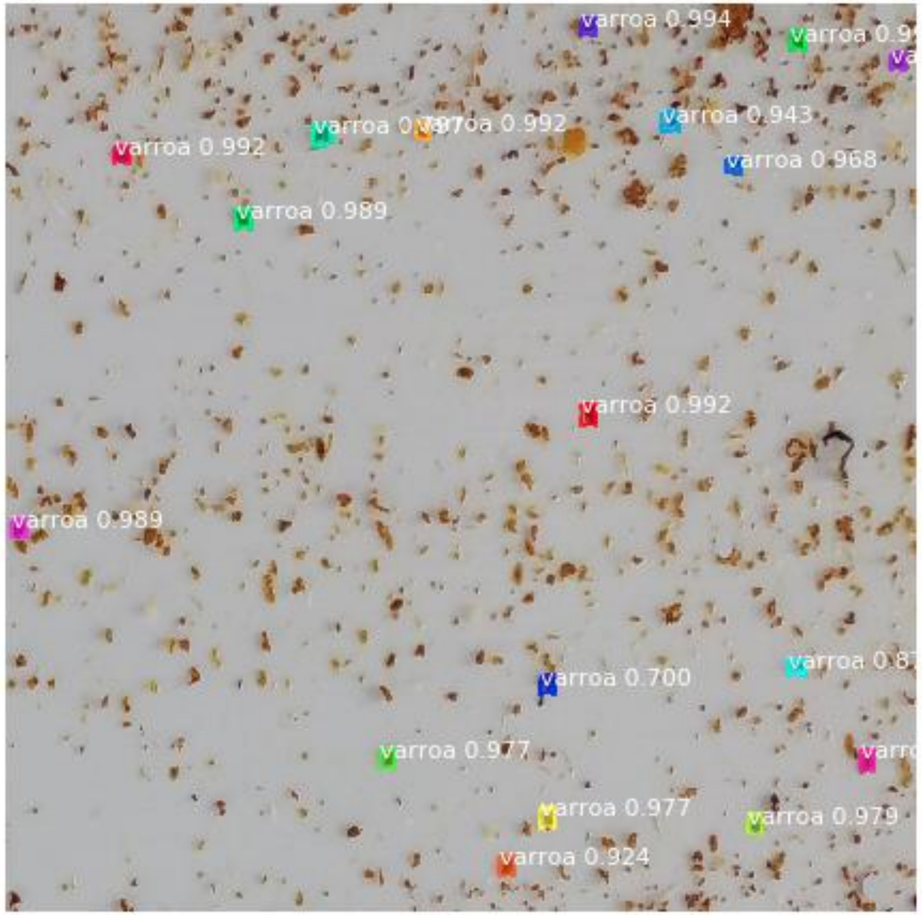
- Hardware:
  - Tesla K80 15G provide by Google Colab
  - Memory: 15G (Quite limited)
- Training:
  - Load pre-trained weight based on coco dataset
  - Load modules from the repo of Matterport Inc.
  - Resnet101(feature extraction) + head(classification)
  - 20 epochs on first 400 training images with 200 steps per epoch
  - 15 epochs on last 400 training images
  - The whole training took about 4-5 hours

MLP and CNN Method – Result: Sample 1

GroundTruth

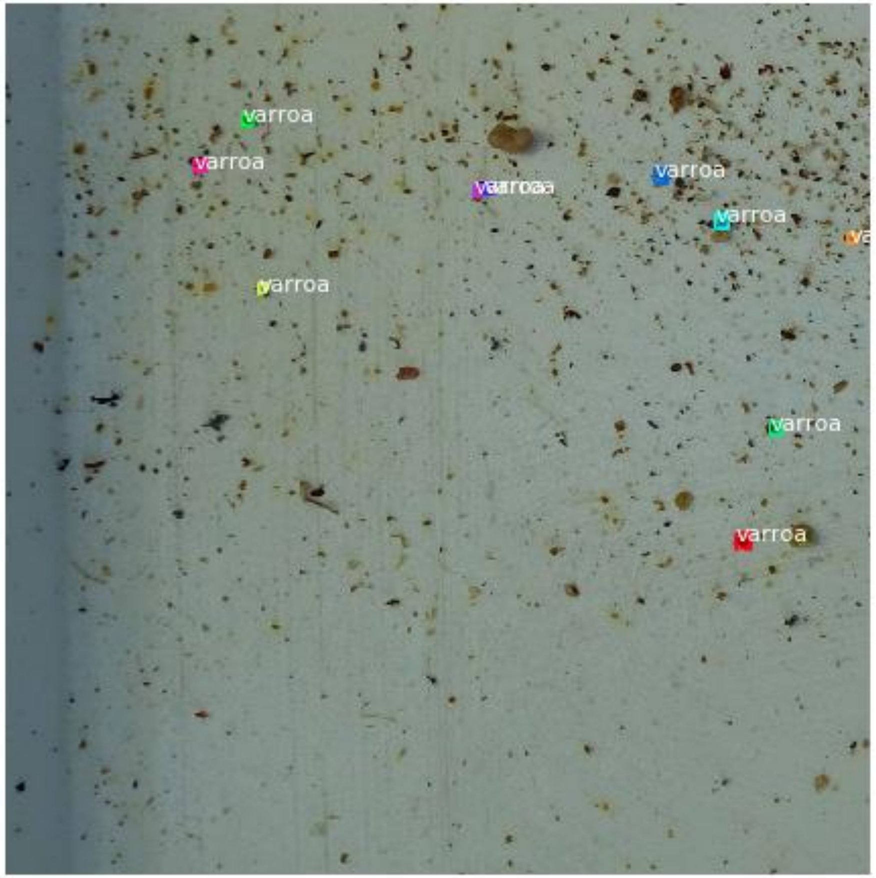


Our result

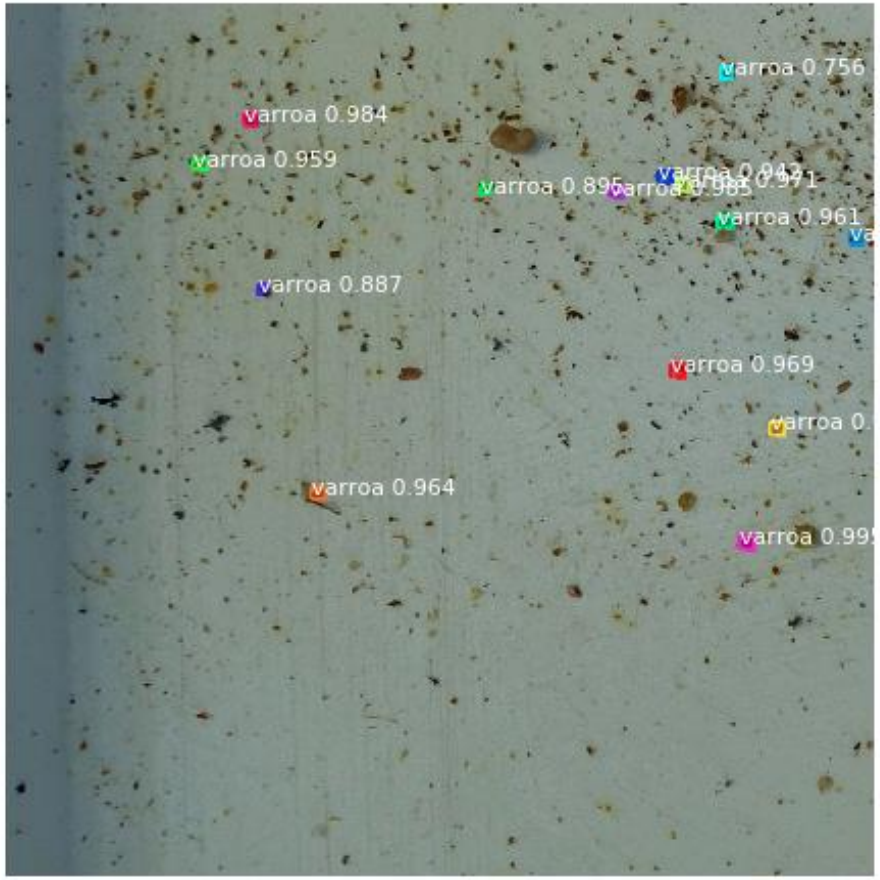


MLP and CNN Method – Result: Sample II

GroundTruth



Our result



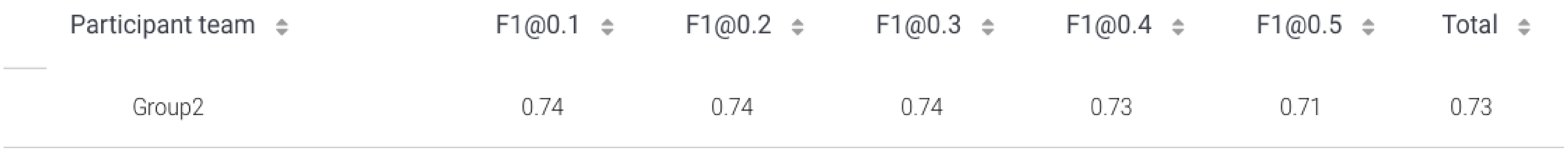
Result

Report of the result on the test @IOU=0.3

- detection by segmentation: 0.29
- detection by sliding window and classifier: 0.23
- detection by Mask-RCNN: 0.71

Report of the result on the competition

- Best F1 score: 0.734



Participant team ⬍	F1@0.1 ⬍	F1@0.2 ⬍	F1@0.3 ⬍	F1@0.4 ⬍	F1@0.5 ⬍	Total ⬍
Group2	0.74	0.74	0.74	0.73	0.71	0.73

Reference

- Image Segmentation with Watershed Algorithm, OpenCV 3.1 documentation
- Rich feature hierarchies for accurate object detection and semantic segmentation. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, CVPR, 2014.
- Fast R-CNN. Ross Girshick, ICCV, 2015.
- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, NIPS, 2015.
- Mask R-CNN, Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, Facebook AI Research (FAIR), 2018
- Mask R-CNN implementation, matterport Inc: https://github.com/matterport/Mask_RCNN

Thank you for your attention!