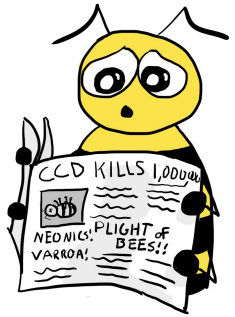
Image Analysis and Pattern Recognition

Final Project
“Varroa Discovery”



Part 1

Find Varroas by Segmentation

Marker-based image segmentation using watershed algorithm

- Basic idea: “Any grayscale image can be viewed as a topographic surface where high intensity denotes peaks and hills while low intensity denotes valleys.*

  What we did?:

- Labelled the region of *sure foreground* or *object* with one color
- Label the region of *sure background* or *non-object* with another color
- The region which we are not sure of anything, label it with 0 → which is our marker

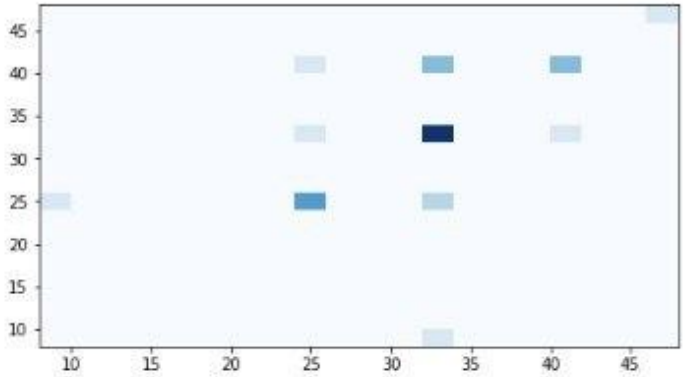
- After applying watershed algorithm our marker will be updated with the labels we gave,
  - and the boundaries of objects will have a value of -1

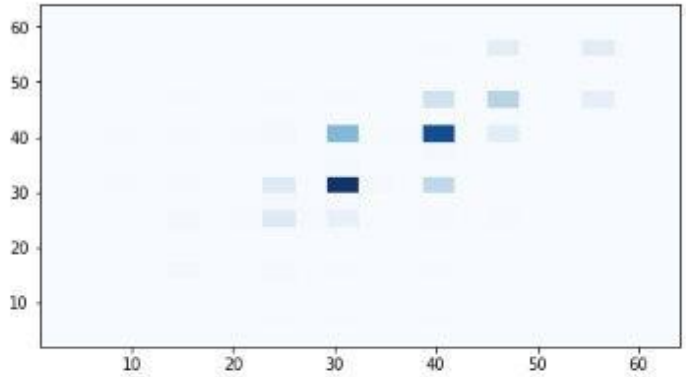
* https://docs.opencv.org/3.4/d3/db4/tutorial_py_watershed.html

Inspect Data

- Choosing a desired area size of the varroas

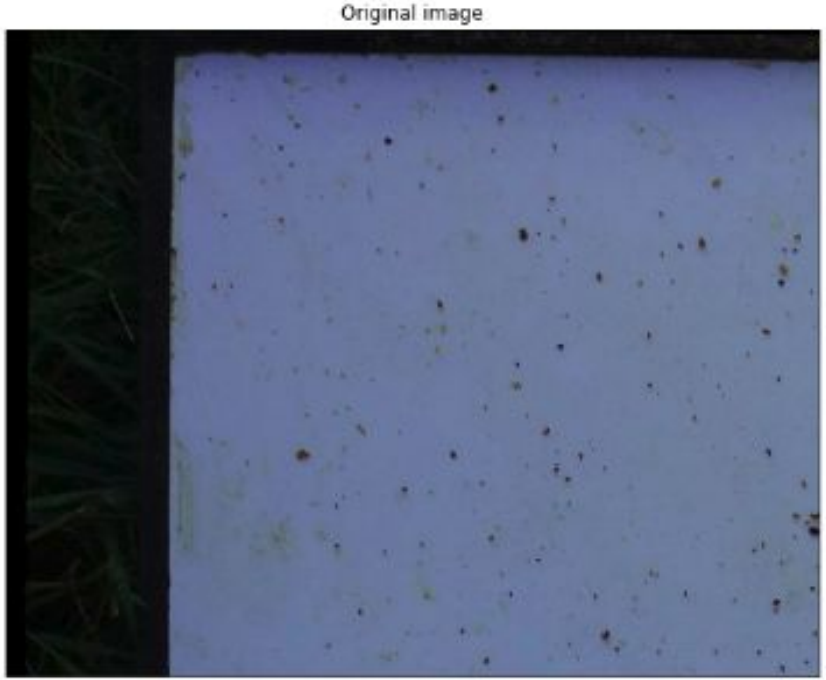
```
Image Area <= 512**2
  Total Varroa:  25
  Varroa Height. mean: 31.68  median: 32.00  min: 8.00  max: 48.00
  Varroa Width.  mean: 31.08  median: 32.00  min: 8.00  max: 48.00
  Varroa Area.   mean: 1017.28  median: 1024.00  min: 192.00  max: 2304.00

Image Area <= 1000**2
  Total Varroa:  9542
  Varroa Height. mean: 37.40  median: 40.00  min: 2.00  max: 64.00
  Varroa Width.  mean: 36.53  median: 40.00  min: 1.00  max: 64.00
  Varroa Area.   mean: 1413.05  median: 1280.00  min: 28.00  max: 4096.00
```

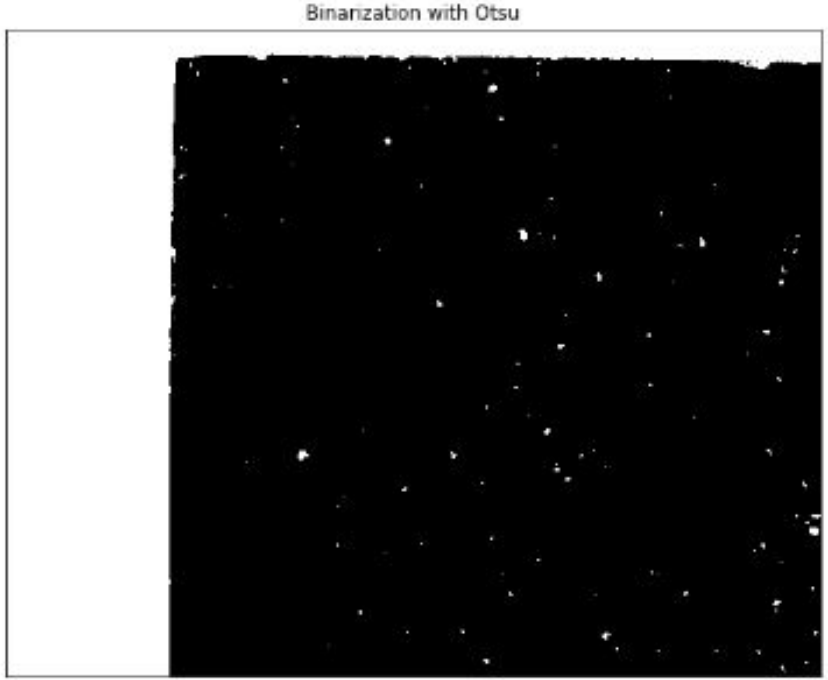




Segmentation Example 1

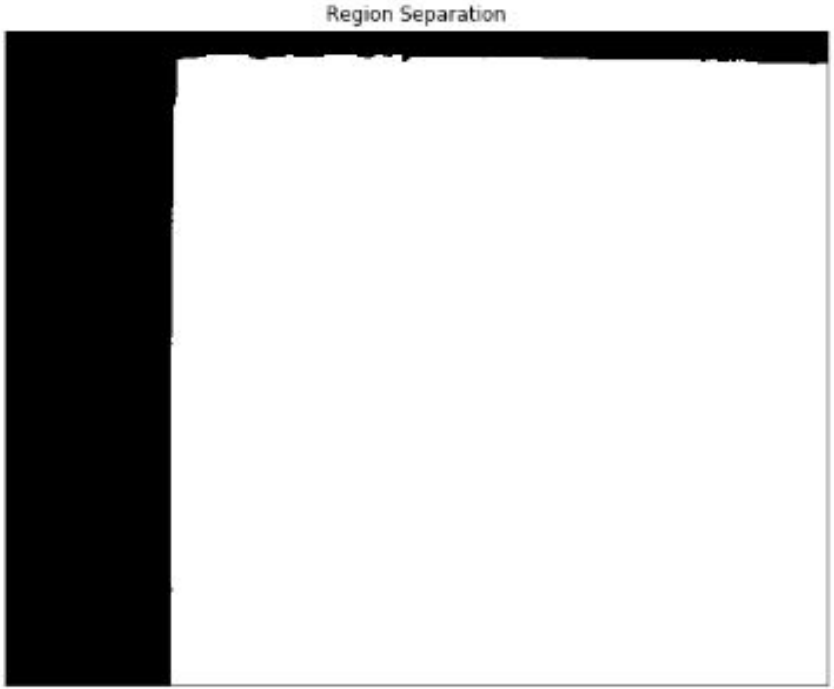
- Finding an approximate of the varroa by Otsu’s binarization

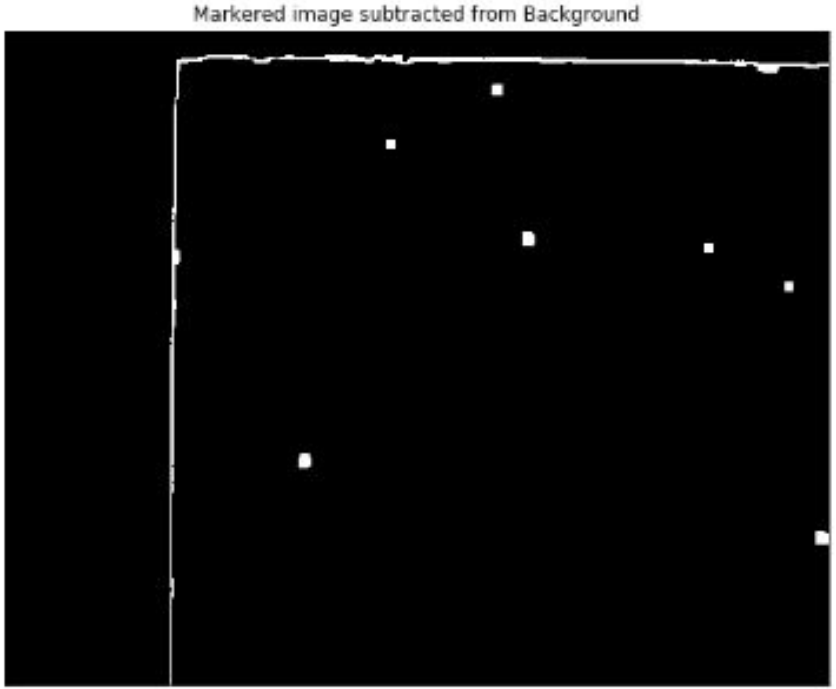




Segmentation Example 1

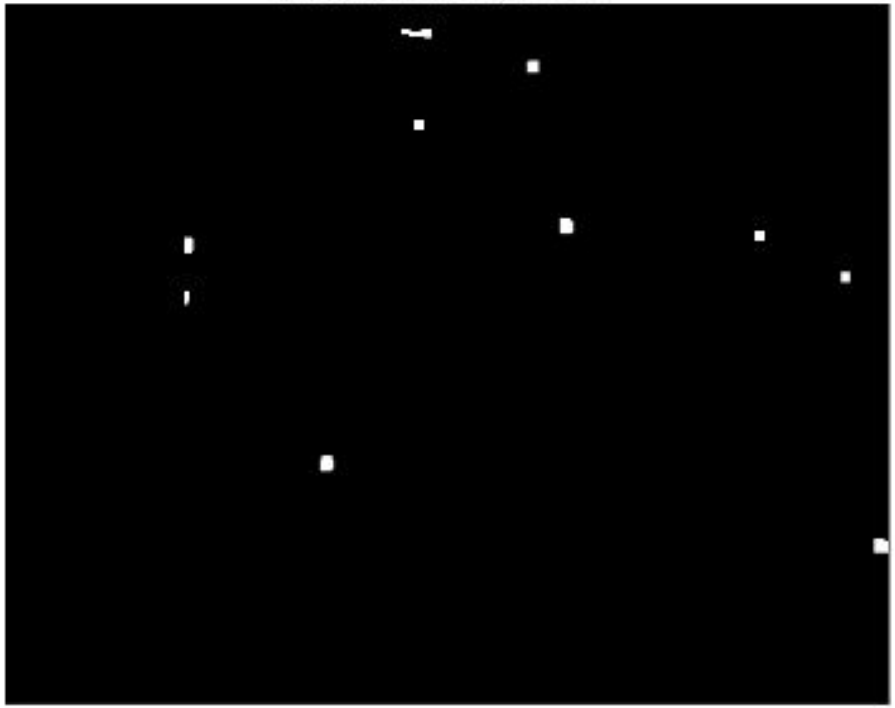
- Mark the regions in different color, ang get subtracted image from background

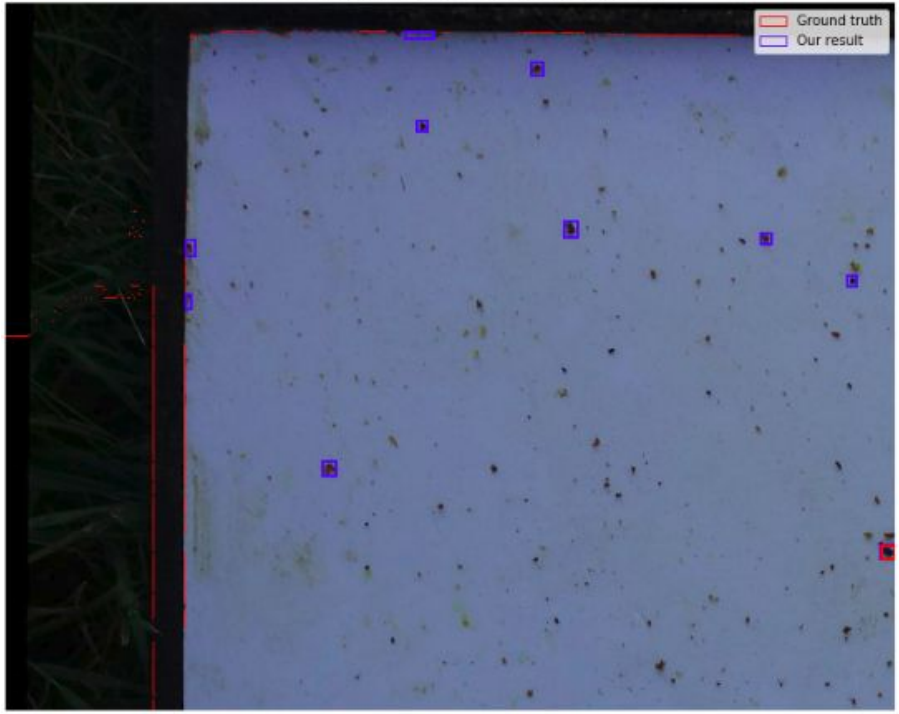




Segmentation Example 1

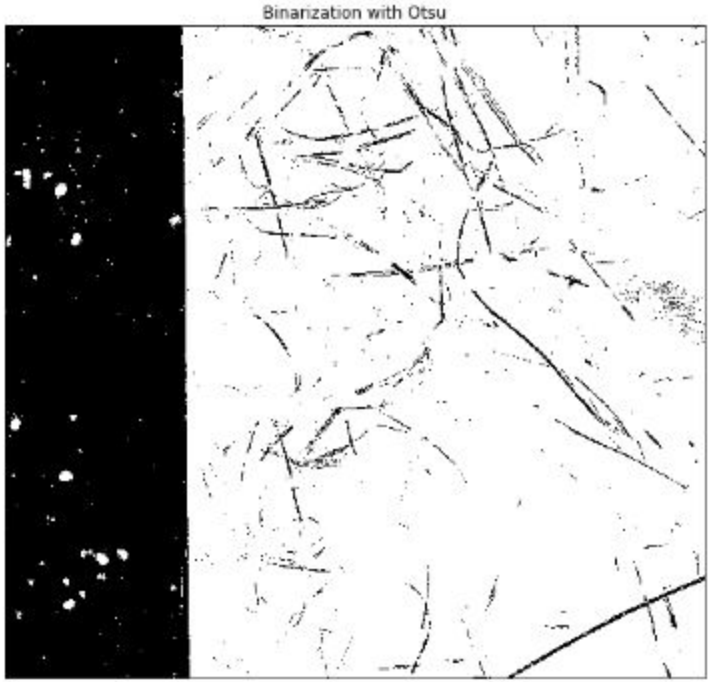
area_max = 1000
area_min = 300





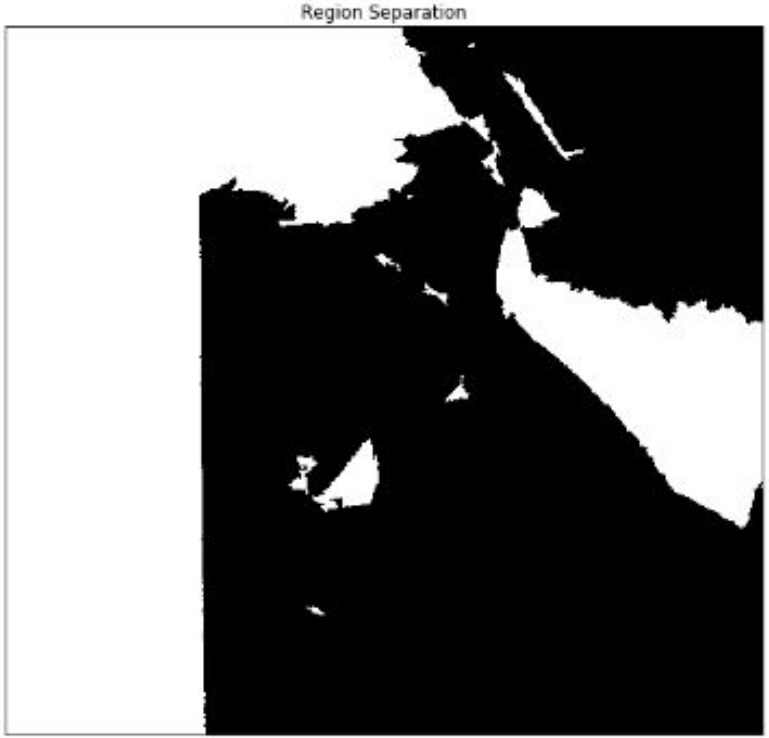
Segmentation Example 2

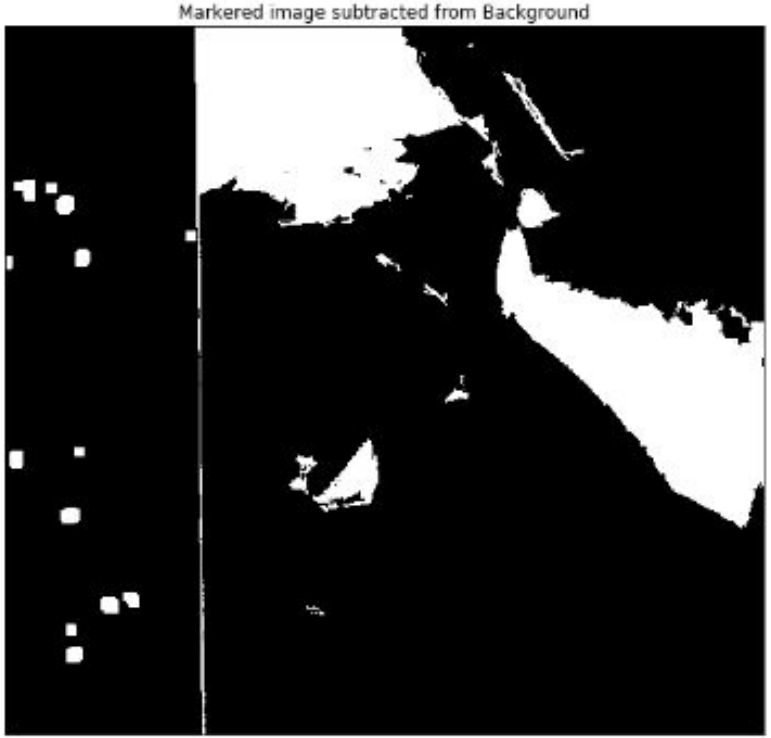
- Finding an approximate of the varroa by Otsu’s binarization



Segmentation Example 2

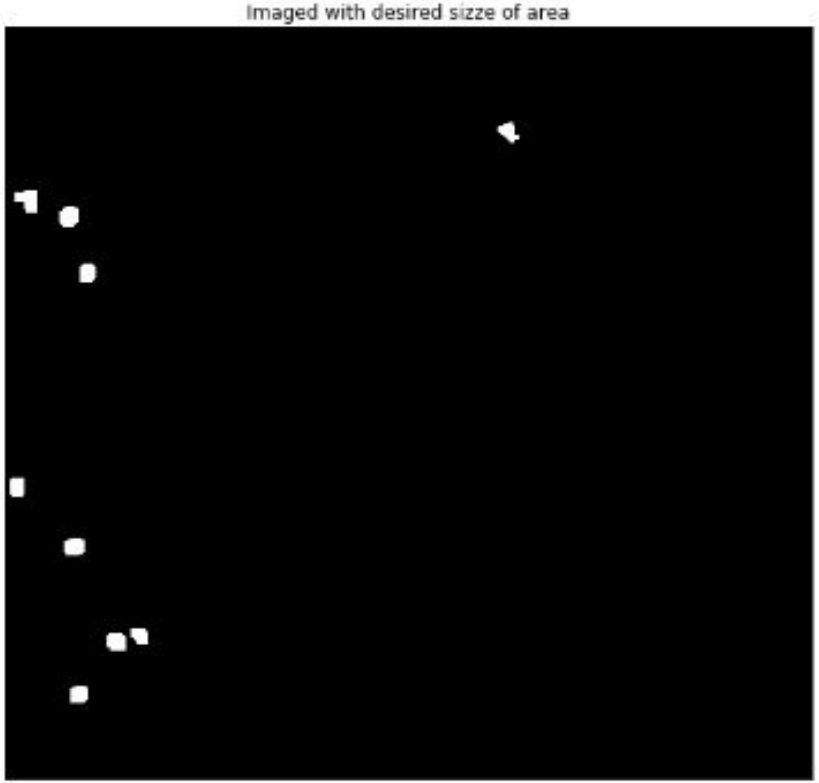
- Mark the regions in different color, ang get subtracted image from background





Segmentation Example 2

area_max = 2300
area_min = 1000

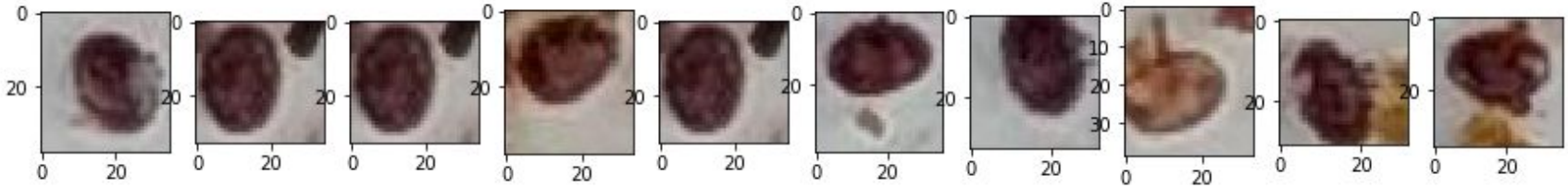


Part 2

Finding Varroas by Object Detection

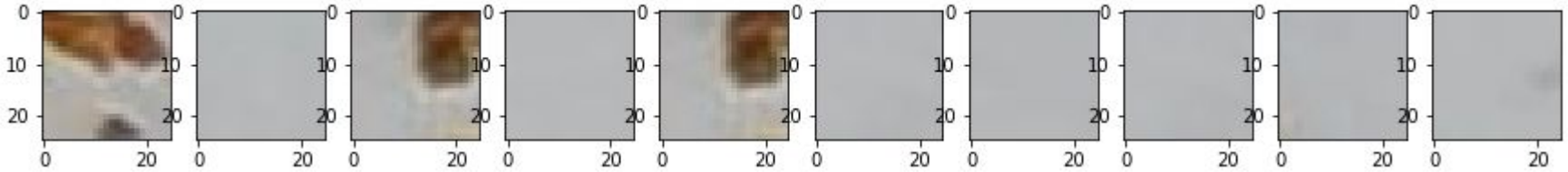
Generating dataset

Varroa samples

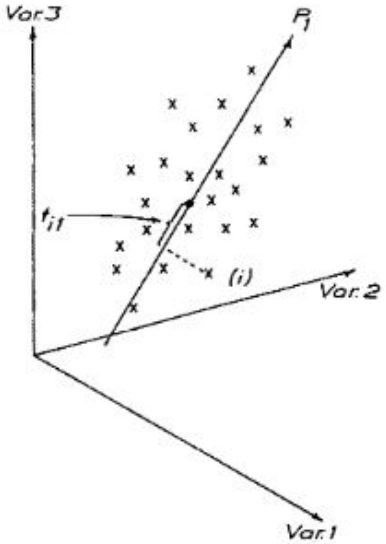


Background samples

- window size chosen by taking a mean of given bound boxes (25x25)
- taking random samples that don’t overlap bounding boxes
- number of background and varroa samples is about the same

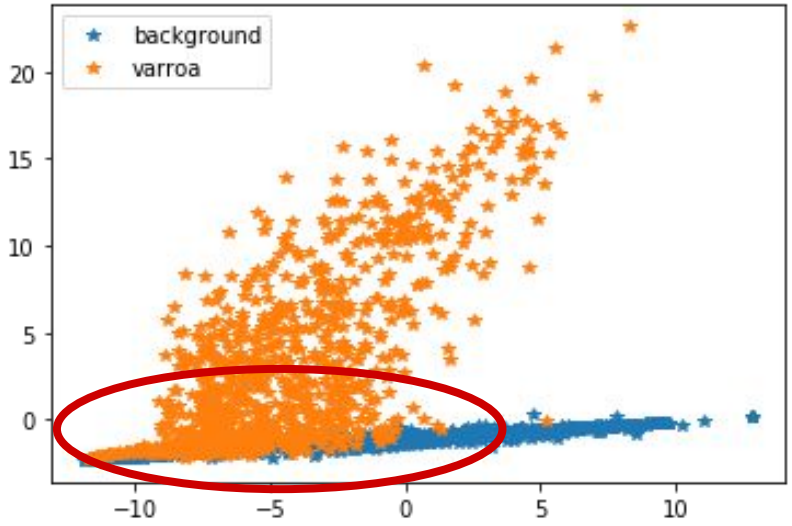


PCA (Principal Component Analysis)

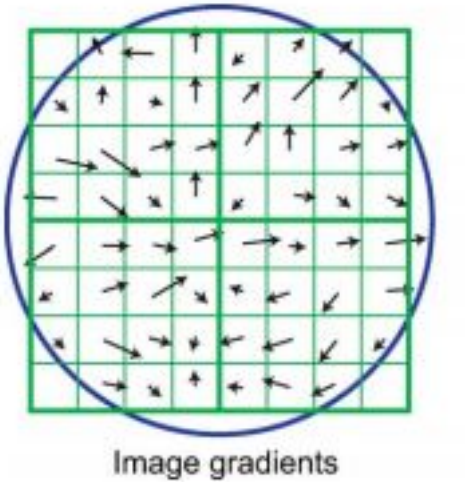


Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.

- Each image is represented as **N**x**M**x**3** feature vector and using PCA reduced to **n** features (e.g. 2, 3)



SIFT (Distinctive Image Features from Scale-Invariant Keypoints)



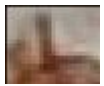
- Family of Histograms of Oriented Gradients (HOG)
- **x** patches for a single window, each patch consist of **128** features => **x * 128** feature for each window
- Number of features has to be reduced!

Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

SIFT: Reducing number of features

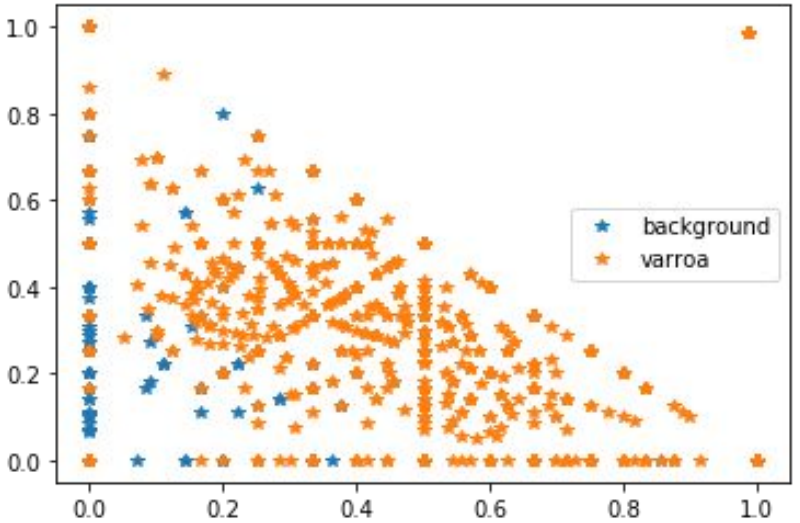
- Consider each point as a SIFT feature (128 dimensional space)
- Cluster points in **k** clusters (e.g. 2, 3)
- Describe each image as a histogram of clustered patches
- Example:

varroa part



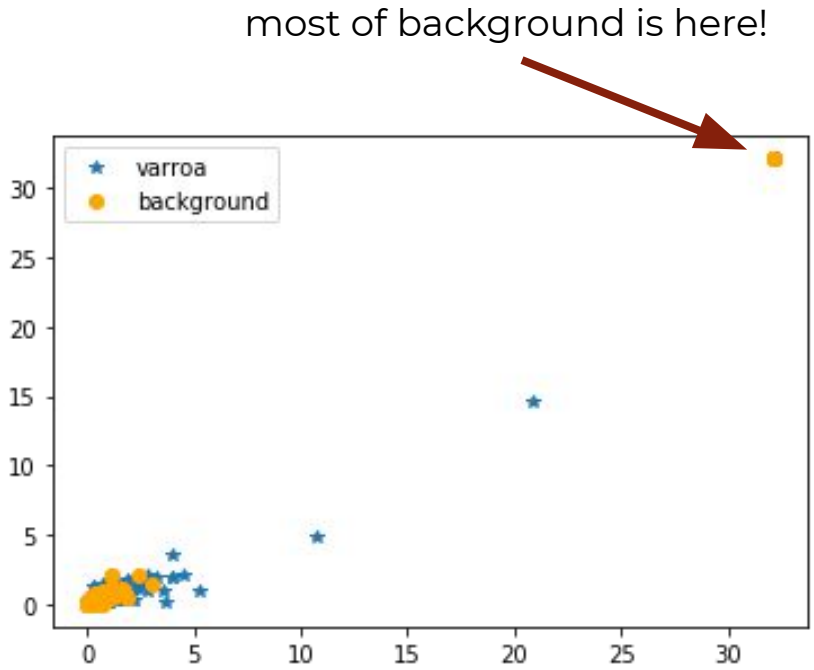
varroa part

background part

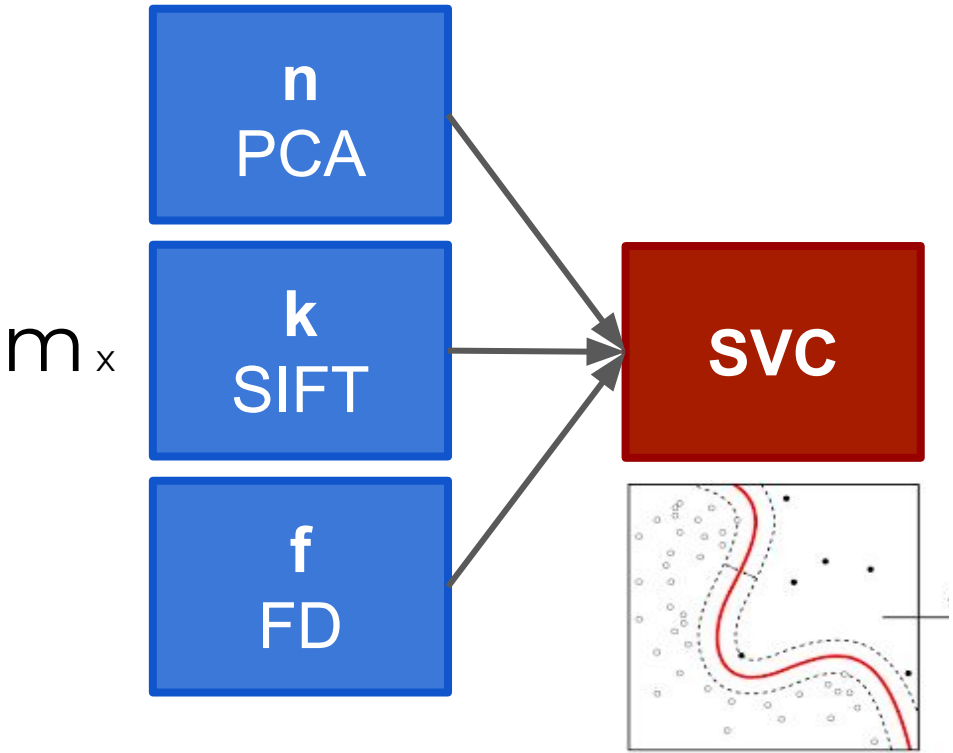


Fourier Descriptors

- Region growing to convert image to binary (varroa vs background)
- Create a contour around the varroa
- Encode the contour for Fourier Descriptors
  - **u_k = x_k + j y_k**
  - **F = FFT(u)**
  - **F_1 is ignored** (translation invariant)
  - **abs(F_3...f) / abs(F_2)** are used to describe the varroa (abs(F_k) for rotation invariant and 1/F_2 for scaling invariant)



SVC (Support Vector Classifier)



- Multiple features (e.g. 7) are extracted from each windows and put in classifier
- **C** & **gamma** parameters of SVC are obtained using Grid Search

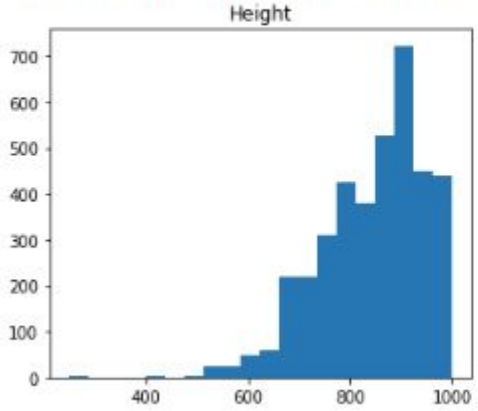
Part 3

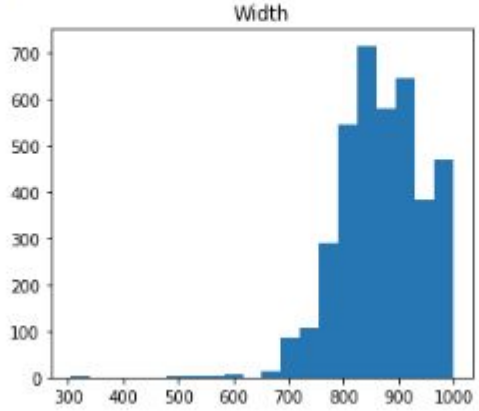
Finding Varroas using Mask-RCNN

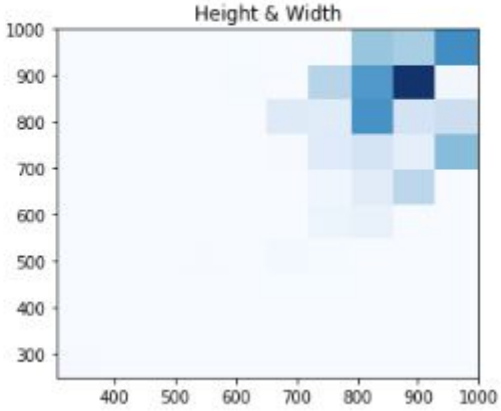
Inspect Data

- Inspect data information

```
Image Count:  3850
Height  mean: 846.22  median: 872.00  min: 248.00  max: 999.00
Width   mean: 869.85  median: 870.00  min: 305.00  max: 999.00
Color   mean (RGB): 138.68 138.68 138.68
```

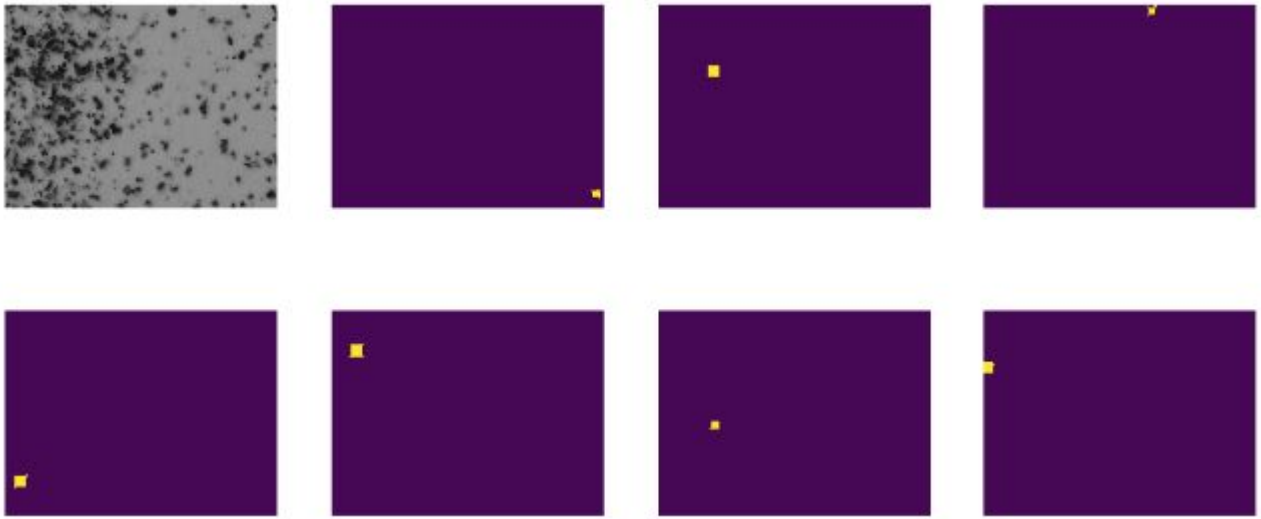






Inspect Data

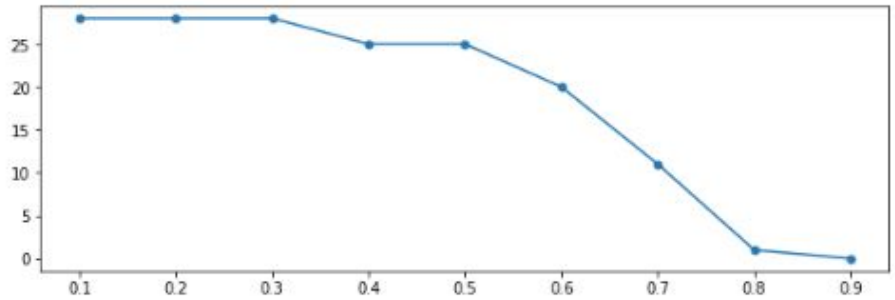
- Split images into 4 subimages
- Generate new data masks



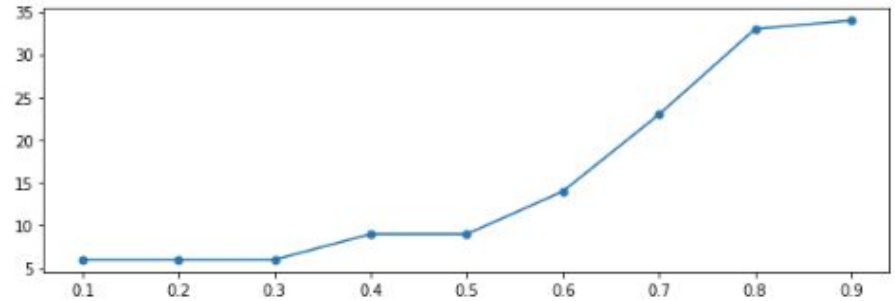
Training

- Mask-RCNN framework [1]
- Train the model
- Tune training hyper-parameters
- Tune score threshold

True Positives w.r.t. Score Threshold



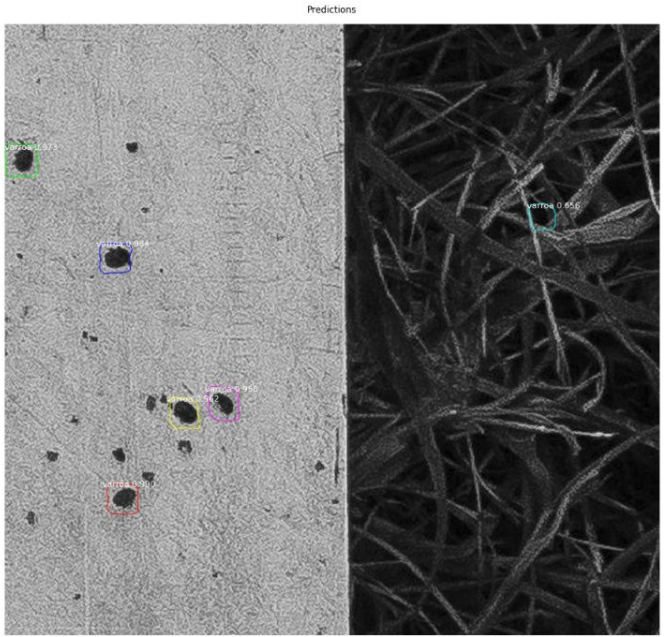
False Positives w.r.t. Score Threshold

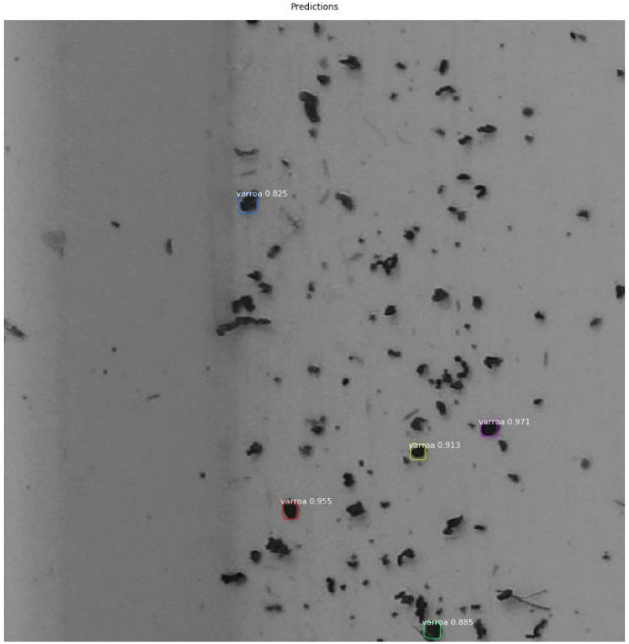


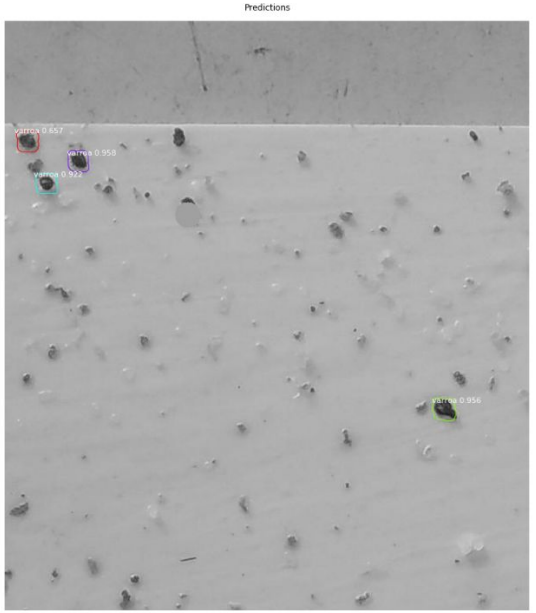
Mask-RCNN framework paper: https://arxiv.org/abs/1703.06870

Merging

- Unite images and their predictions
- Submit JSON file with RoIs







Results

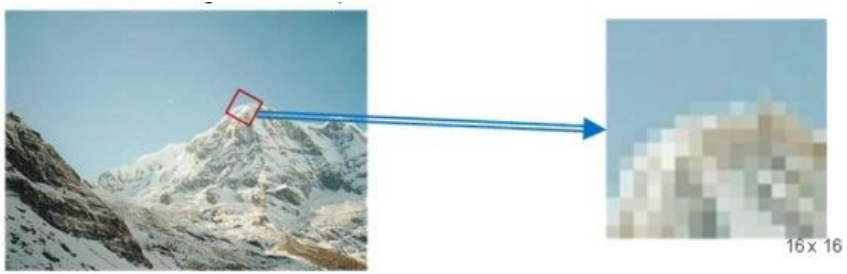
	F1@0.1	F1@0.2	F1@0.3	F1@0.4	F1@0.5	Total
20 last layer; 40 all layers	0.74	0.74	0.74	0.73	0.71	0.73
20 last layer; 48 all layers Score threshold 0.9	0.67	0.67	0.66	0.65	0.64	0.66
20 last layer; 48 all layers Score threshold 0.8	0.58	0.57	0.56	0.56	0.55	0.57
20 last layer; 40 all layers Less data	0.47	0.46	0.45	0.45	0.44	0.46

Appendix

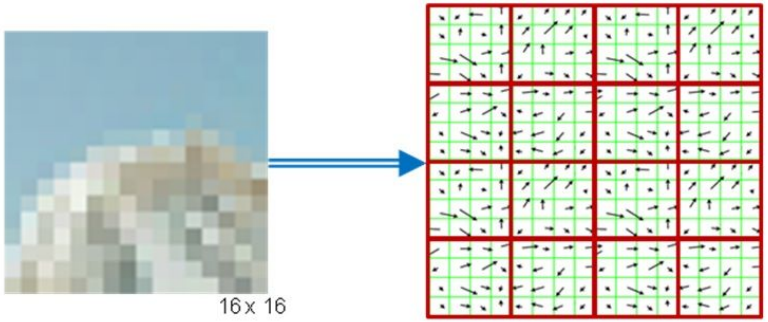
Additional materials

SIFT

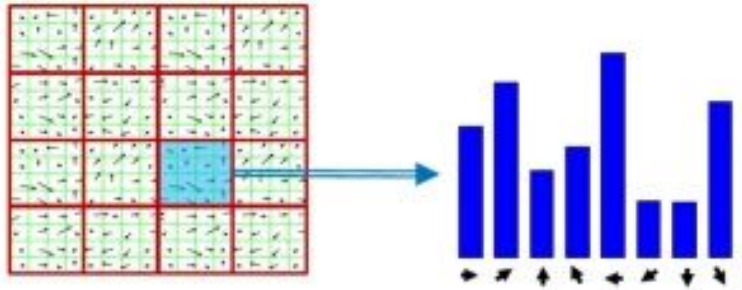
1.



2.



3.



4.

5.

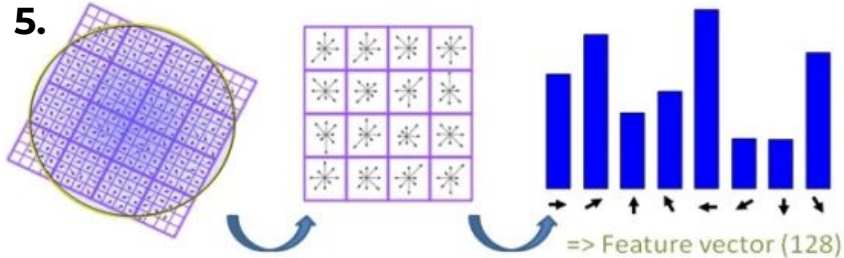
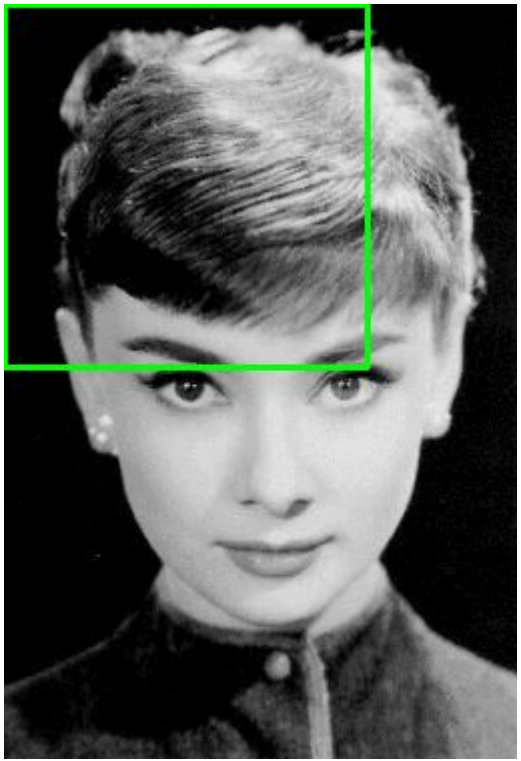


Image from:
https://gilscvblog.com/2013/08/18/a-short-introduction-to-descriptors/

SVC (Grid Search & Results)

```
SVC(C=250, cache_size=200, class_weight='balanced', coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
  max_iter=-1, probability=True, random_state=None, shrinking=True,
  tol=0.001, verbose=False)

Classification report
              precision    recall  f1-score   support

  background       0.98      0.98      0.98      1361
      varroa       0.96      0.96      0.96       582

   micro avg       0.98      0.98      0.98      1943
   macro avg       0.97      0.97      0.97      1943
weighted avg       0.98      0.98      0.98      1943
```

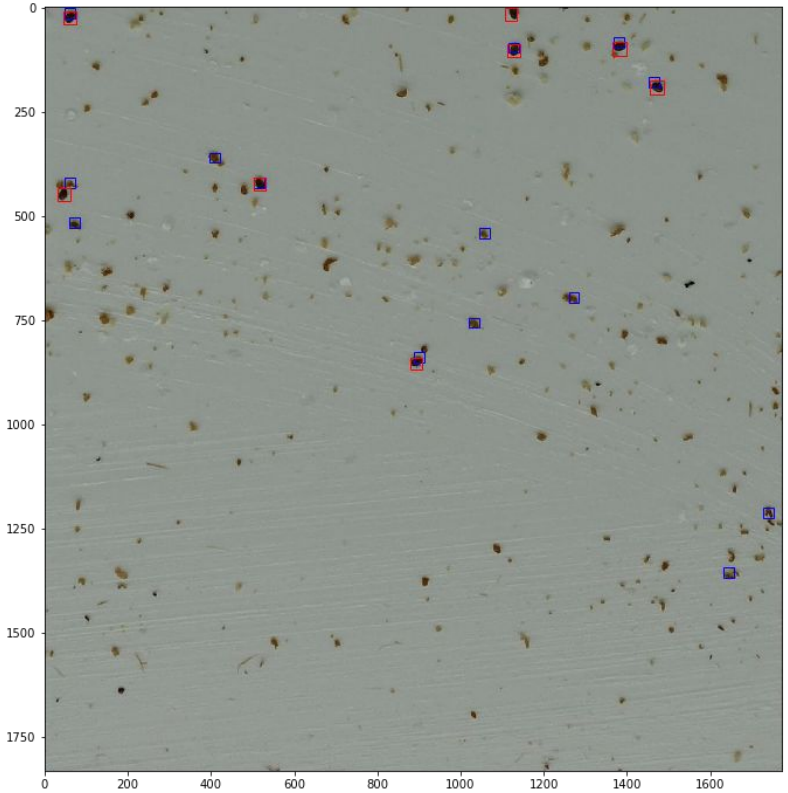
Sliding Window



- Window size: 25 x 25px
- Window overlap: 50% (along x & y axis)

Image from:
https://www.pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/

Description: Example of classification



Blue: Our Classification
Red: Ground Truth