Automatic Signature Stability Analysis and Verification using Local Features

Team - Morphors

TA Mentor : Sowmyasis

Faculty Mentor : Ravi kiran

Repository:
https://github.com/Digital-Image-Processing-IIITH/project-morphers

Project ID: 6

Raja - 2018102032

Mohith - 2018102016

Tejaswini - 2018102018

Gowthami - 2018102048

Main Goals of the Project

1. To have a **fully automatic system** that can classify signatures as **genuine, forged, disguised** provided a few reference signatures are available in database based on the implementation of the given paper

2. Achieve an **equal error rate of <= 15%** on the **4NSigComp2010** dataset, the most well known publicly available dataset of **forensic signature verification** competition

Uniqueness of the paper

- ➔ The system is fully **offline**
- ➔ Using local **SURF** features to perform stability analysis
- ➔ Achieves a **low Equal Error Rate(EER)** even in the presence of **DISGUISE Signatures**
- ➔ **Equal Error Rate of 15%** on 4NSigComp2010 Dataset. Best team in the competition achieved an **EER of 55%**

Uniqueness of what we did

- We studied upon **SURF, SIFT, ORB** and used all these 3, we obtained results which would be later used for comparison and analysis.
- We have used simple techniques to implement the algorithm which was mentioned in the paper.
- Although we have a simple code, the time taken isn't much either.
- We have implemented this code not only on the Dataset given to us but also our signatures and verified the correctness/ efficiency of the algorithm
- We have also tried the process for different values of threshold and the values and have analysed the graphs in the coming slides

Local Stability Analysis(1/2)

- In the case of digital signature analysis, to determine whether the signature is done by same person or not is very important.
- So when signature images are analysed, it is very important to note the major problem here, which is we don't always put the curve or dots at the same angle i.e 2 signatures can't be exactly same in all the pixels.
- So to determine whether if the signature is trustworthy or not, it's important to consider "stable points" where if the images are analysed, the image can be classified as real signature or a forgery.
- So to determine these stable points in the signature(because there are variations) we perform stability analysis

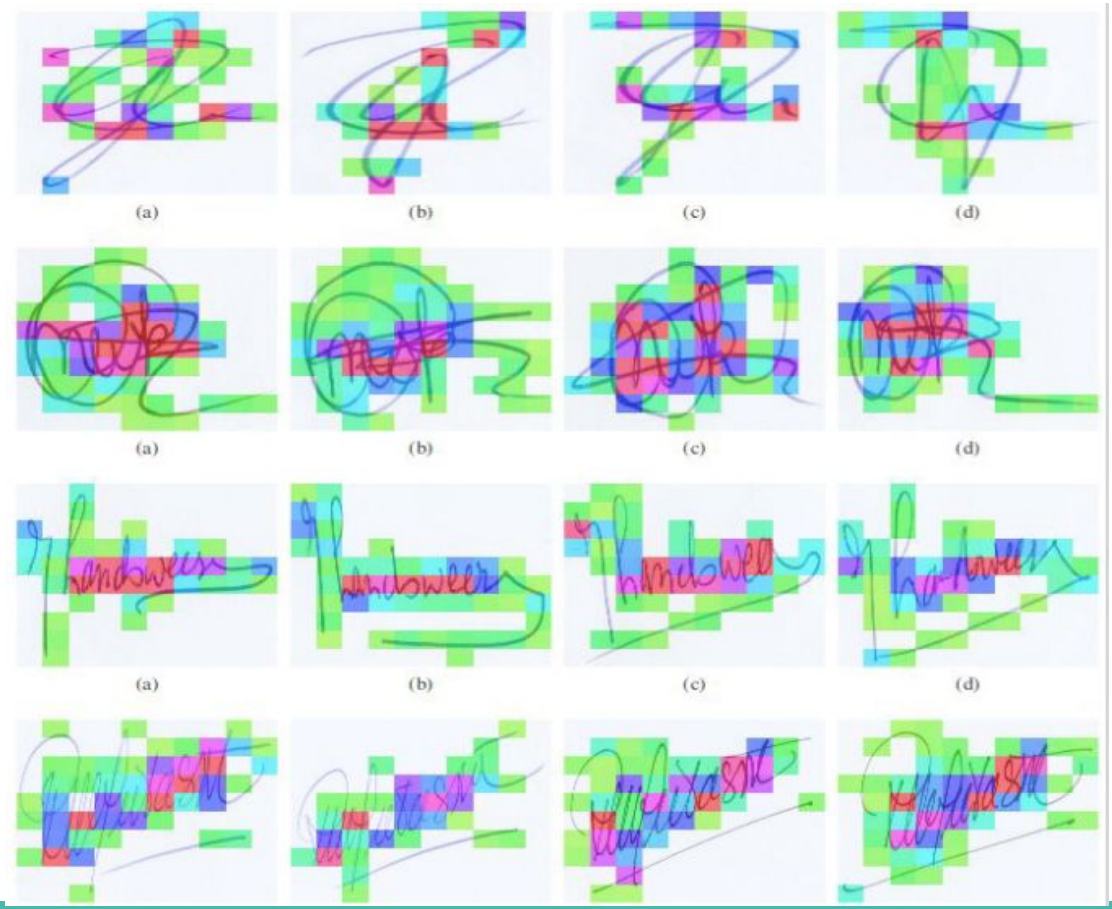
Local Stability Analysis(2/2)

- There are 2 types of stability analysis, global and local.
- Global stability analysis involves taking the whole pictures to analyse
- In case of Local stability analysis, small portions of image are considered by functions are SIFT, SURF, ORB, BRISK etc., and they are analysed to compute stable points
- In our case of signature analysis, Local Stability analysis is better because in case analysis of signatures, the small curve shapes in the signatures have more information about the signature rather than the information we get by analysis of image on the whole.
- So in this paper we use local stability analysis to differentiate between simulated(fake) and disguised(real) signatures.

An example given in the paper is shown in the next slide

An Example to show the stability regions in an image

- As you can observe there are different colored regions in the figure show.
- Here each color represents the type of stability of the particular part of the signature
- Green represents the most stable parts whereas red represents the most unstable parts of the signature



SURF-Introduction(1/2)

- The paper chooses SURF(Speeded up Robust Features) for the analysis of the signatures.
- To be more elaborate, we extract stable features using surf and try to match those points between the reference signatures and the query signatures.
- SURF is a patented local feature detector and descriptor. It is partly inspired by the scale-invariant feature transform SIFT descriptor.
- SURF descriptors have been used to locate and recognize objects, people or faces, to reconstruct 3D scenes, to track objects and to extract points of interest.
- The image is transformed into coordinates and using the multi-resolution pyramid technique, original image is copied so that new image have same shape reduced bandwidth.
- The “stable features” here remain invariant because of which this method is popular to extract features

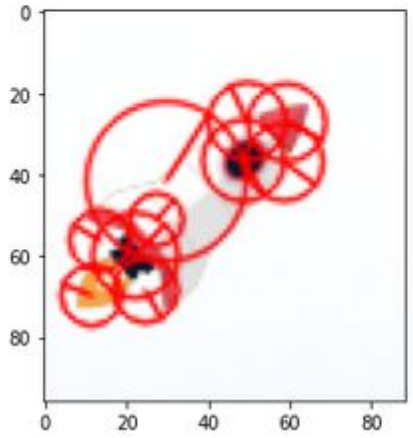
SURF-(2/2)

- For image keypoints and descriptors extraction, as image is considered through image pyramid, the image is smoothed by gaussian filter.
- Surf Algorithm used box filter as an approximation of gaussian filter which helps in making the process much faster if integral image is used.
- As second step during finding the points of interest, SURF uses hessian matrix which has double derivative entry values.
- To assign orientation to a feature, haar wavelet responses are extracted by using small region around the interest region.
- As the final step in the algorithm, we compare different descriptors from different images and matching pairs can be determined.
- In short, SURF adds a lot of features to improve the speed in every step. Analysis shows it is 3 times faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change.

SURF- Example

SURF is performed on the above figure.
As it is observed in the right hand side figure, using the key points circled in red, the image can be constructed again.
Similarly descriptors of this image can be used for determining if 2 pictures are similar by matching them at these key points.

```
img = cv2.imread('rocket4.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
surf = cv2.xfeatures2d.SURF_create(hessianThreshold = 1000, extended = 1)
kp, des = surf.detectAndCompute(img,None)
img2 = cv2.drawKeypoints(img,kp,None,(255,0,0),4)
plt.imshow(img2)
plt.show()
```



Algorithm-(1/2)

- Our Algorithm has mainly 2 functions. One is for creating a reference database and other is to classify the descriptors are matched or not and there return stable points where descriptors match.
- In database function, we compute SURF keypoint descriptors from all reference signatures except one signature and store in a temporary database(tmp_DB) and then compute SURF keypoint descriptors from the remaining signature and find the distance of each keypoint to the nearest descriptor present in tmp_DB.
- Find average distance(d) and eliminate all key points having distance greater than d. Descriptors having distance less than or equal to d are added to final database(DB).
- We repeat this same process in the order of images given in the reference folder to get the final reference database.

Algorithm-(2/2)

- In the classify function, we took the images present in genuine, disguise and simulated separately.
- One by One image is taken and after extraction of its SURF keypoints and descriptors we compare the descriptors of this image and the reference database descriptors.
- We put a threshold using which we determine how many points matched.
  - Minimum distance Less than threshold - stable
  - Minimum distance between threshold to 3*threshold- slightly stable
  - Minimum distance greater than 3*threshold - unstable
- We return the matched percentages for each of the images in genuine, disguise and simulated folders.
- This is done for training set. For test set the same procedure is followed except where classify function is only called on one folder named query signatures and reference dataset is used for creating reference database

Note:- descriptors are 128 dimension vectors

Metrics and parameters

- **False Acceptance Rate(FAR) :** Measure of the likelihood that forged signature will be incorrectly classified as genuine. After we get match percentages for each image, genuine_match and disguised percentages are used to calculate this as they are real signatures and some of the classified to be fake.
- **False Rejection Rate(FRR):** Measure of the likelihood that genuine or disguised signature will be incorrectly classified as forged. Simulated_match percentage is used for this because they are actually fake signatures but some are classified as real.
- **Equal error rate (EER):** When the above error rates are equal, the common value is referred to as the equal error rate
- The parameters like threshold in classify and theta used for calculating FAR, FFR are set after experimenting with different values of them and observing the results.

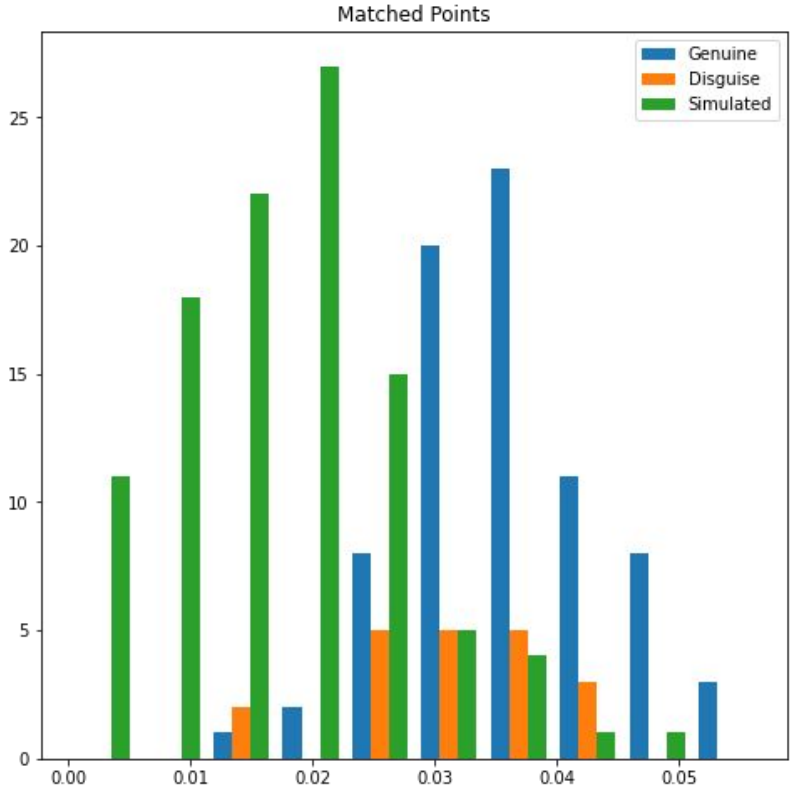
RESULTS

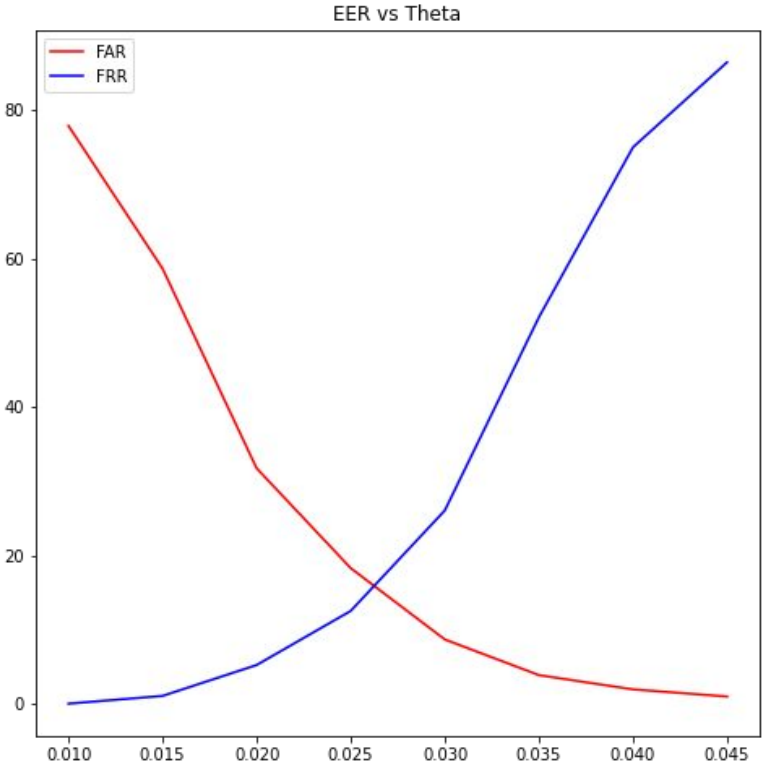
Stable and unstable points of train reference set

Stable and unstable points of test reference set

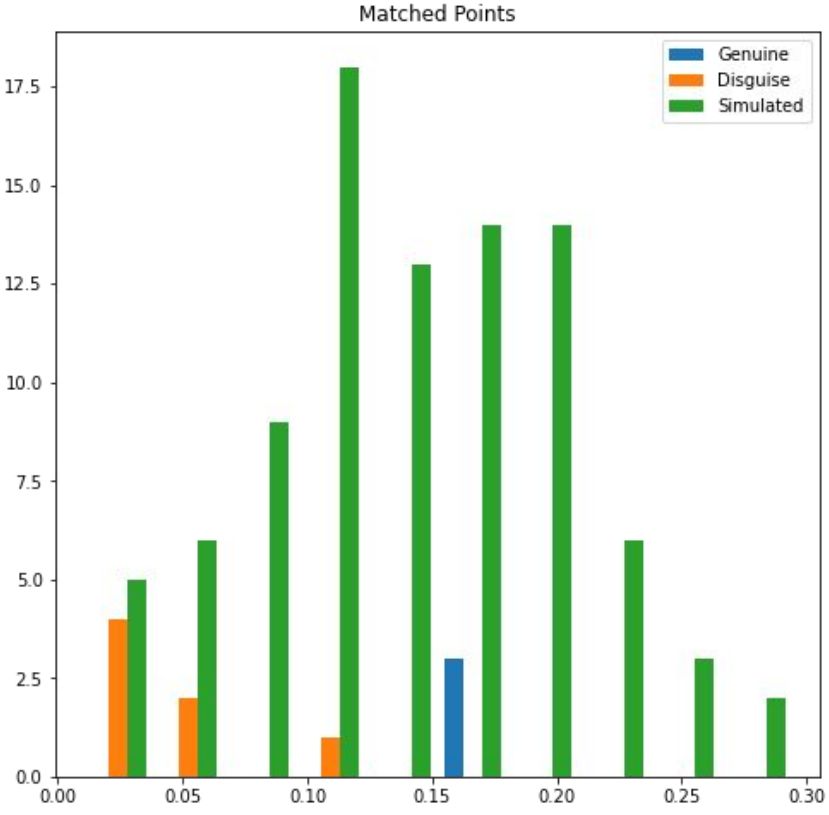
Stable and unstable points for a generated set

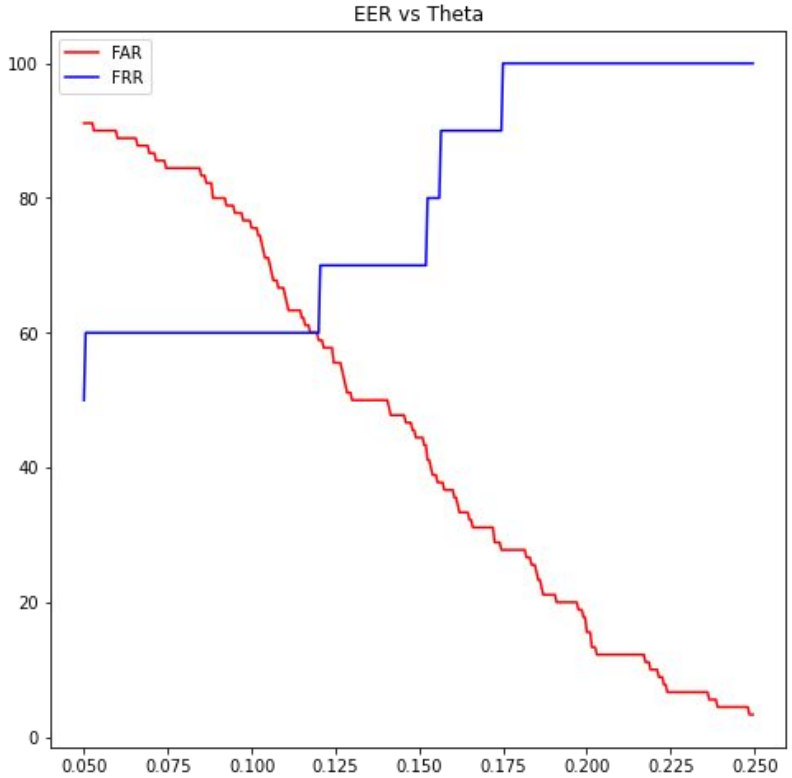
Train set for Thresholding value : 0.11



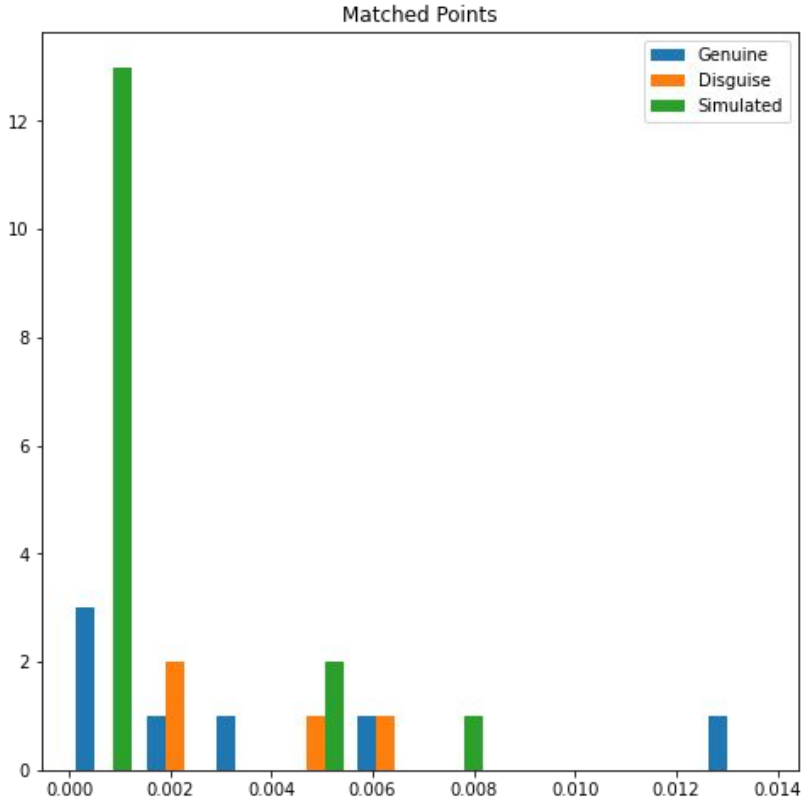


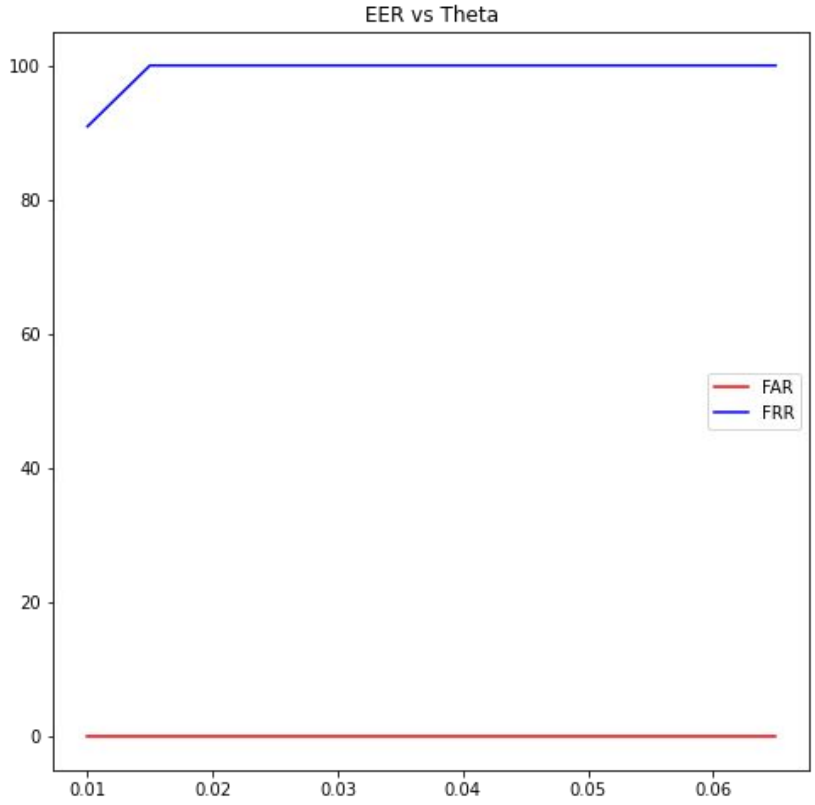
Test set for thresholding =0.11



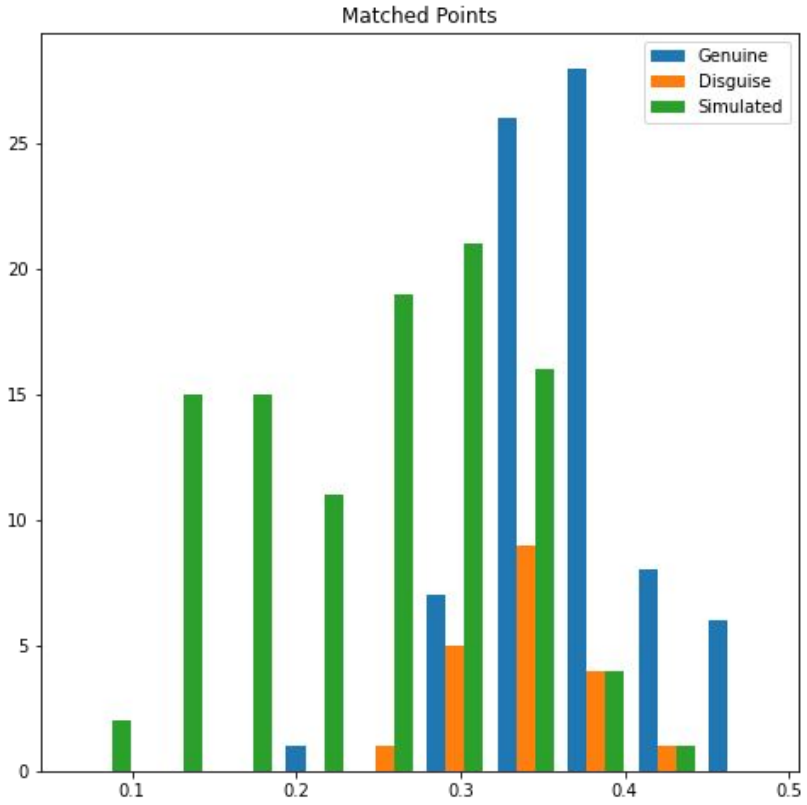


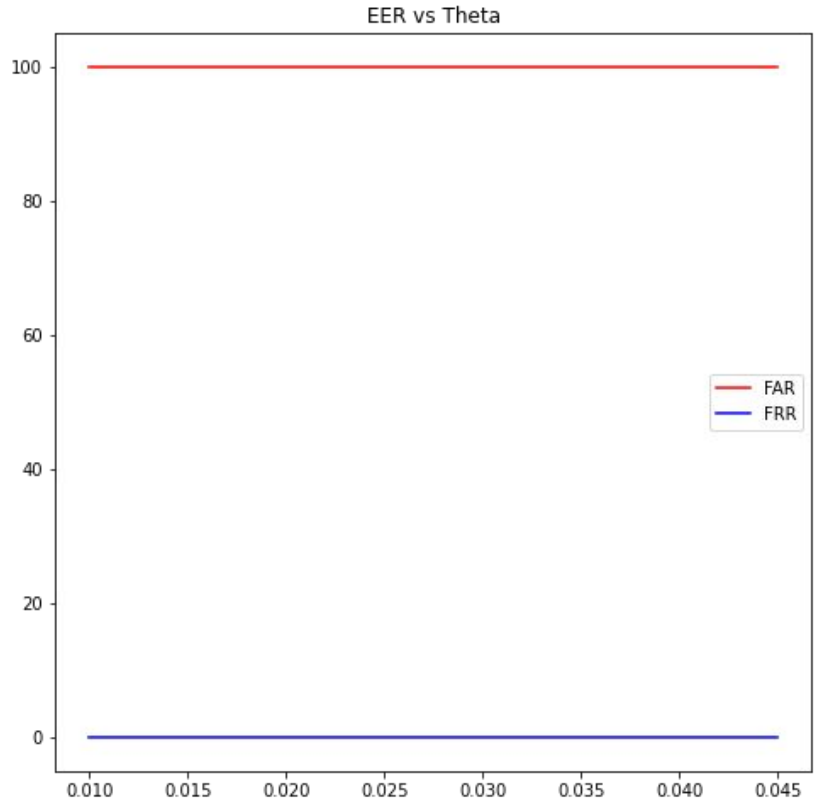
Generated signatures with thresholding = 0.11



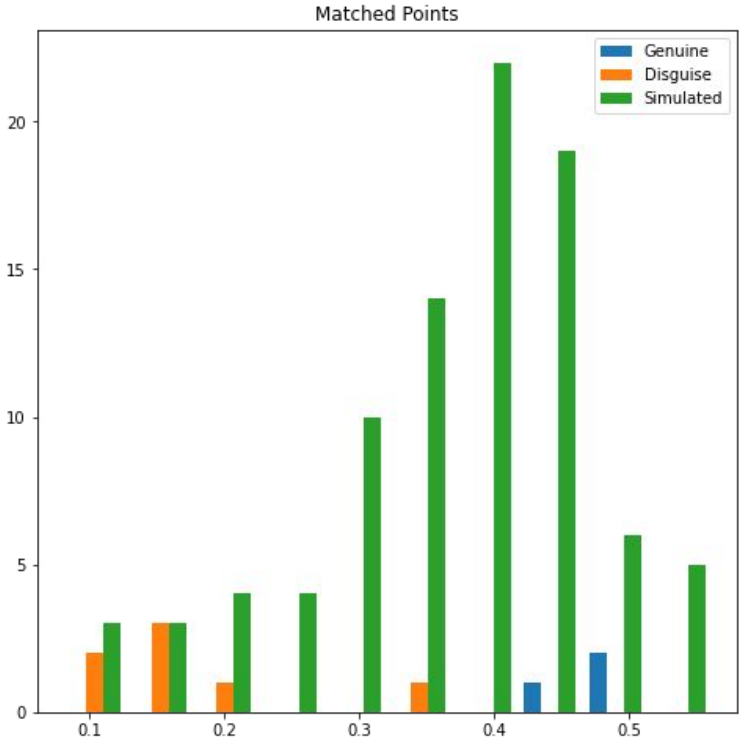


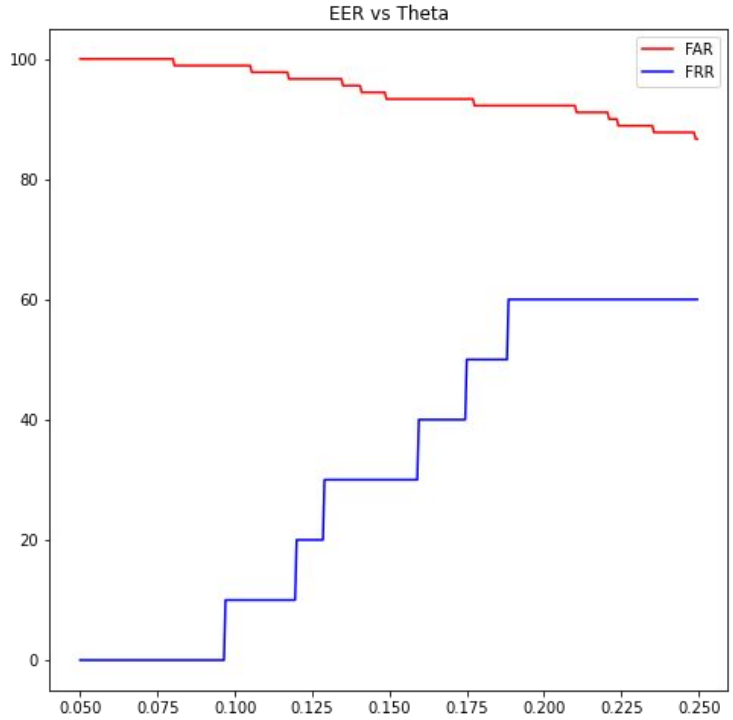
Train set for thresholding = 0.17



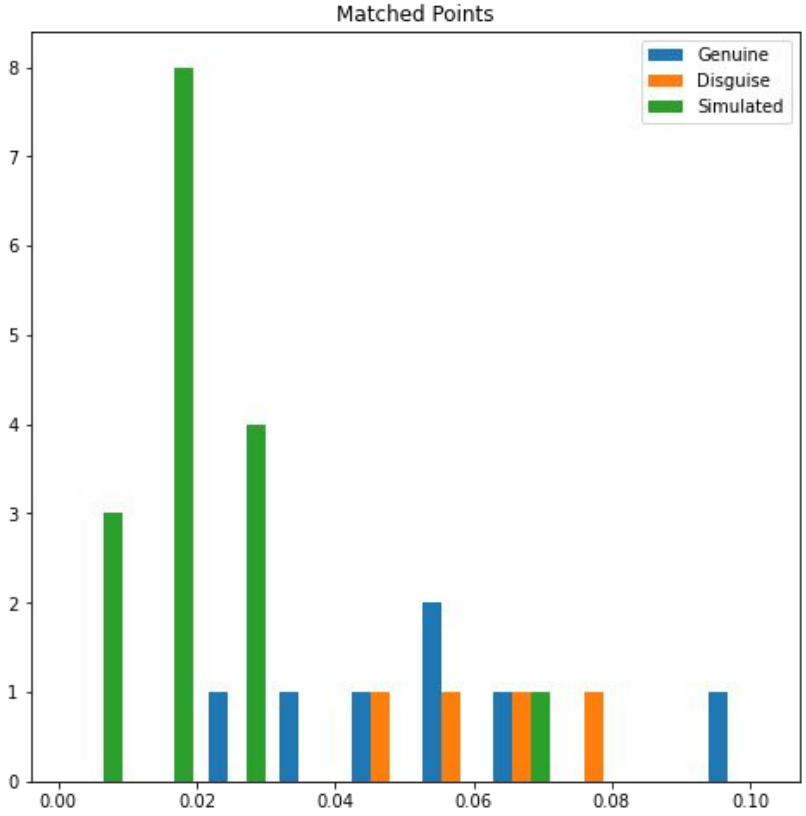


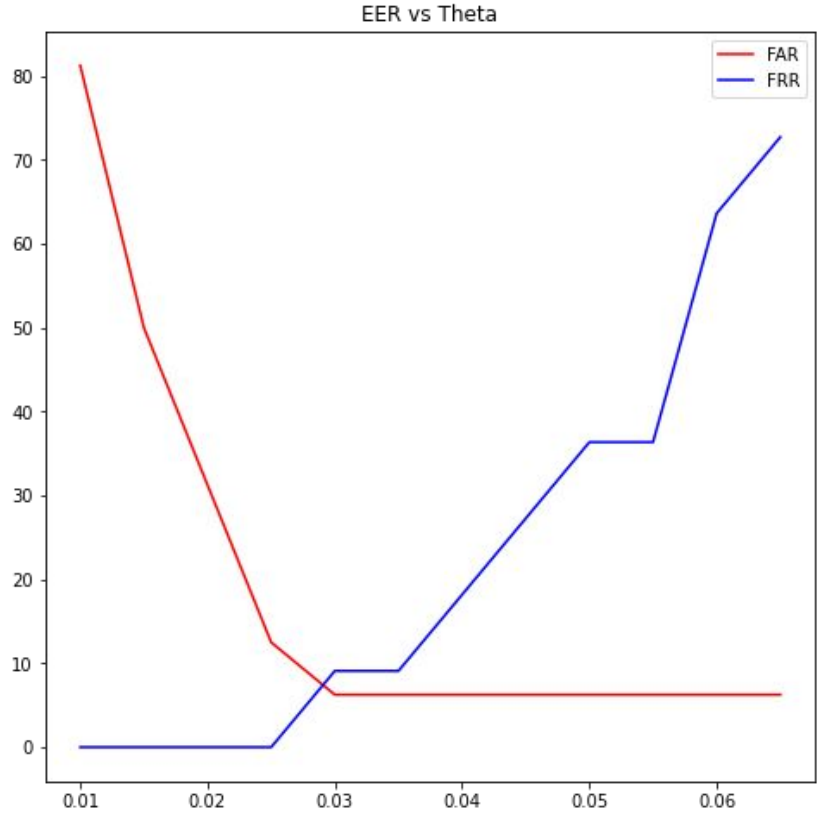
Test set for thresholding = 0.17



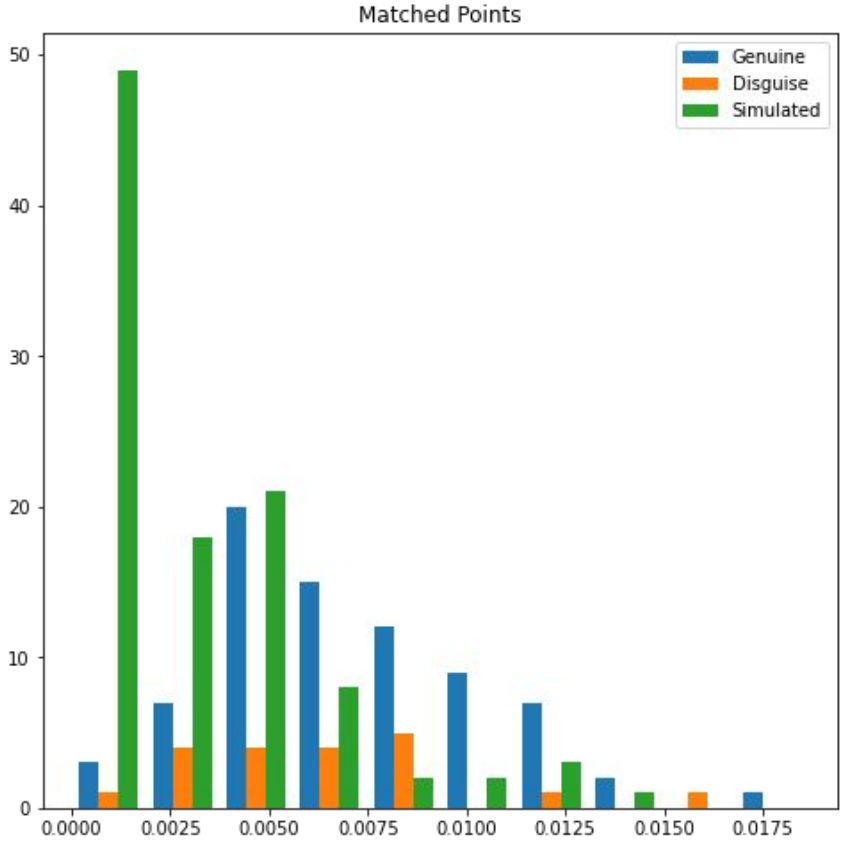


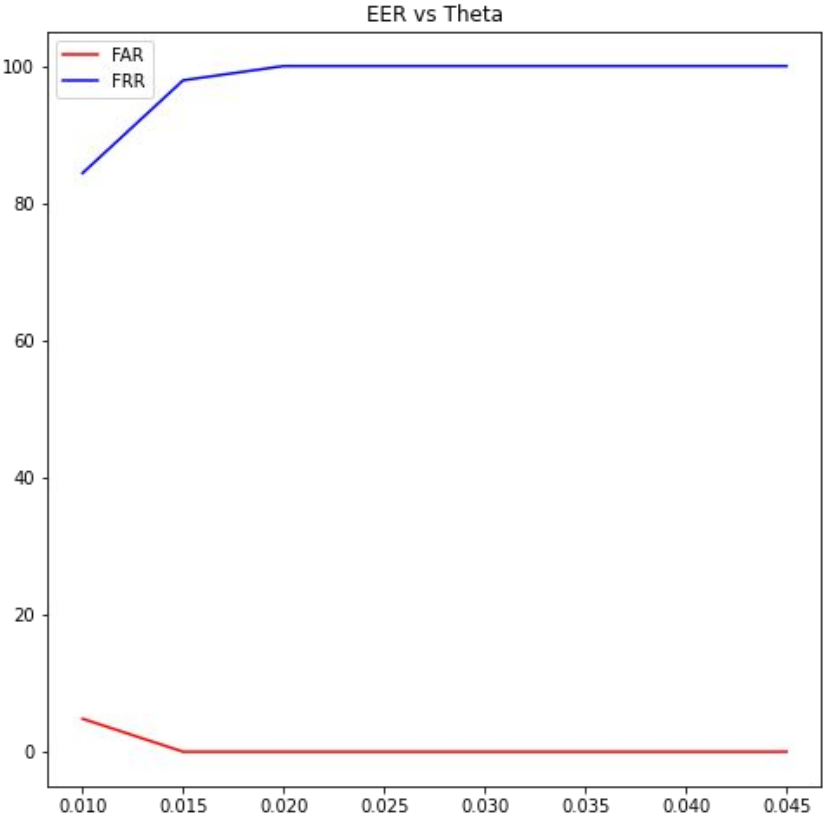
Generated set for thresholding = 0.17



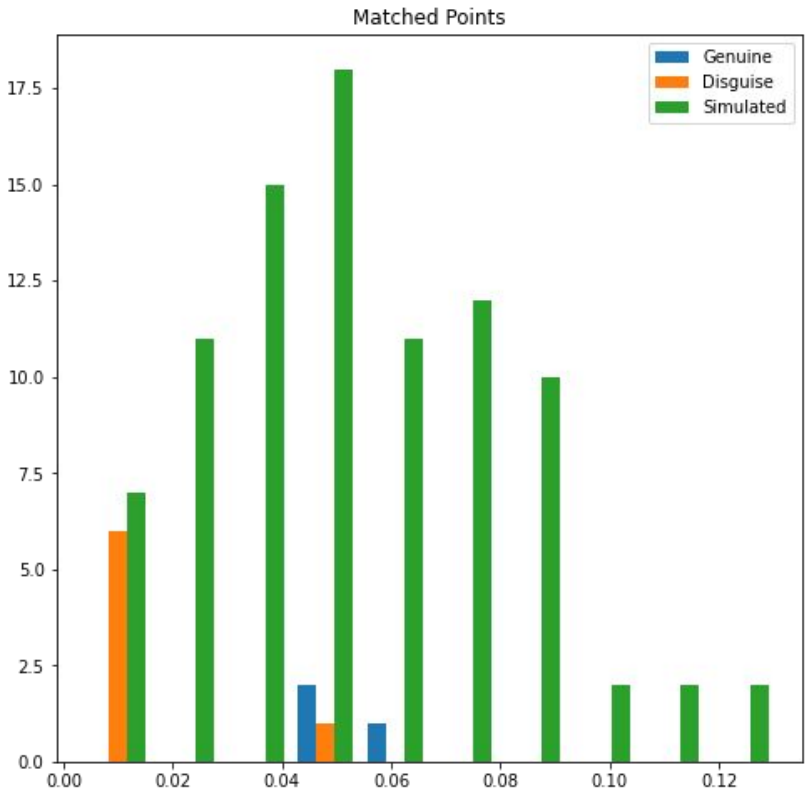


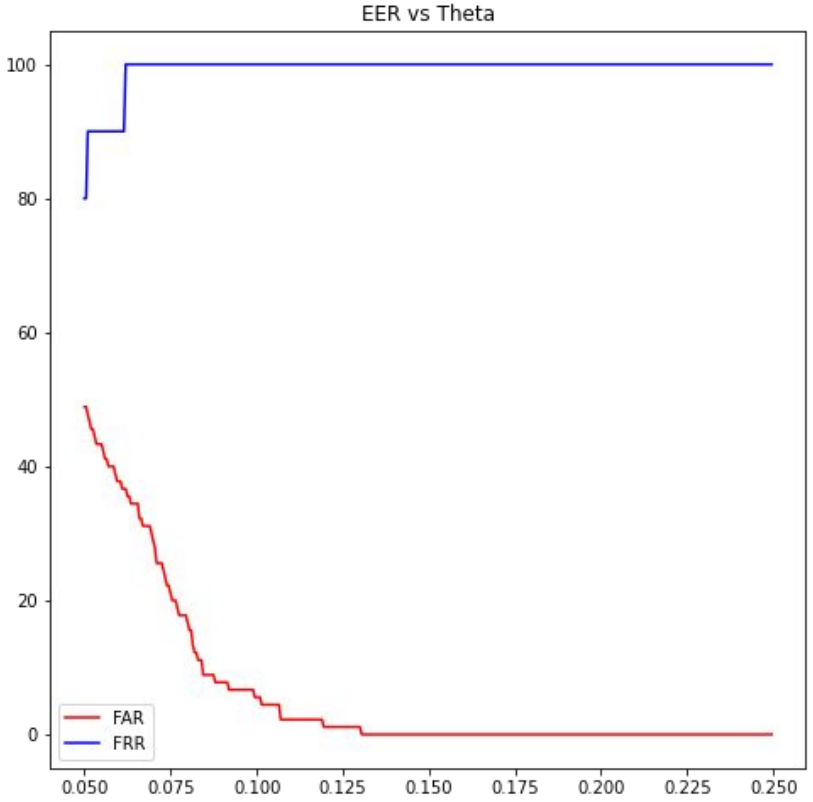
Train set for threshold=0.09



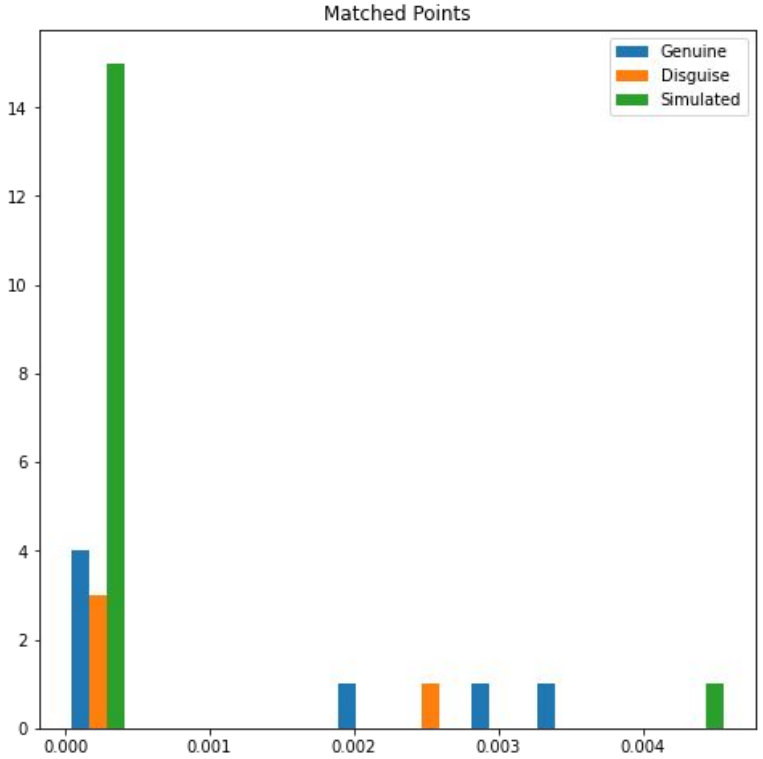


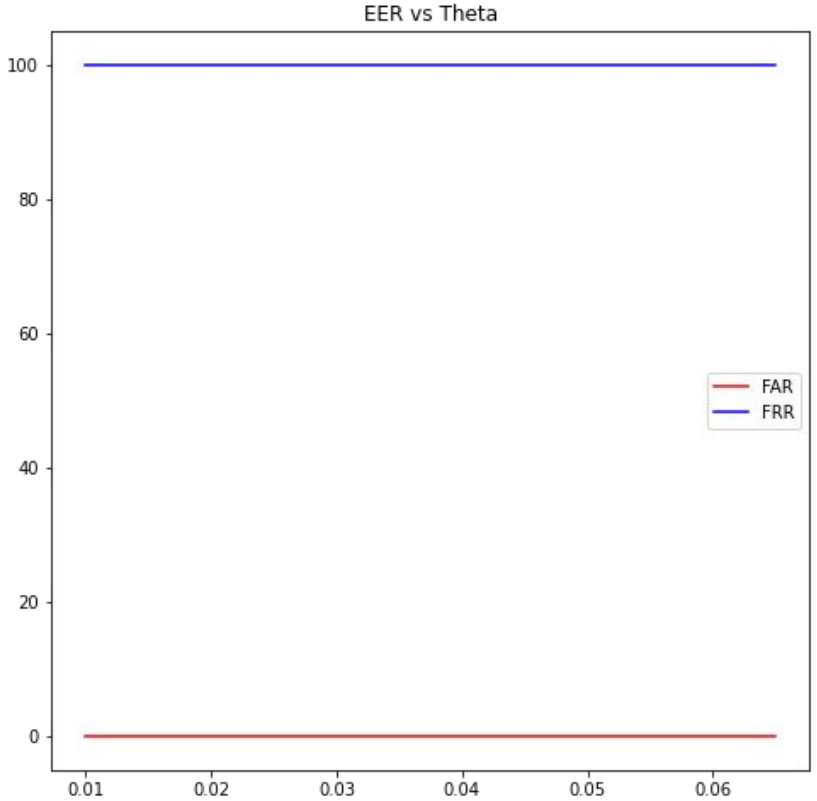
Test set for thresholding = 0.09



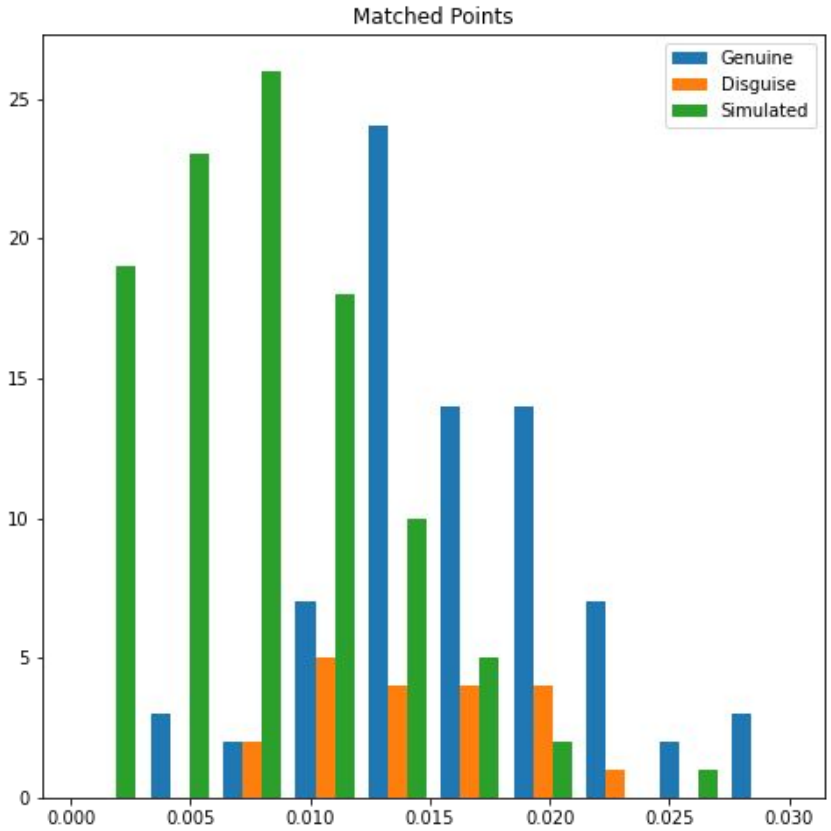


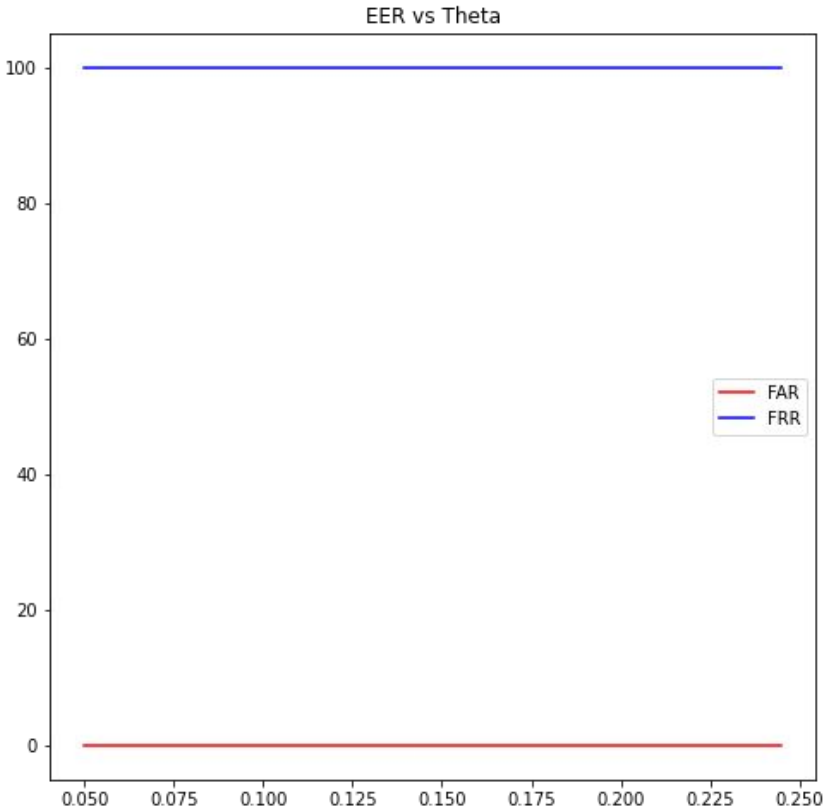
Generated set for threshold=0.09



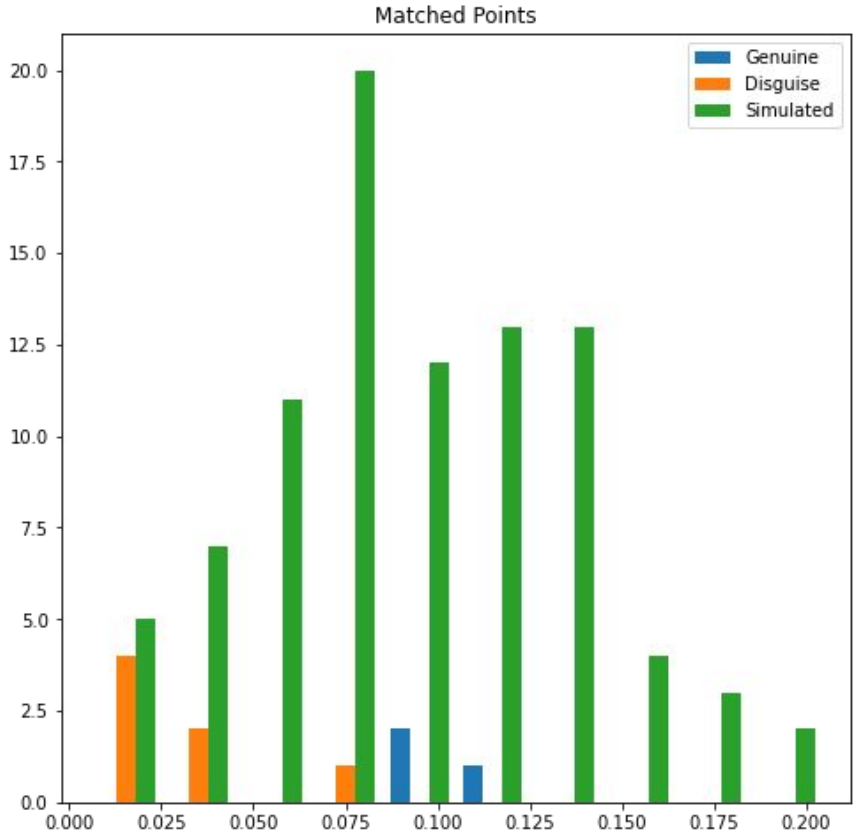


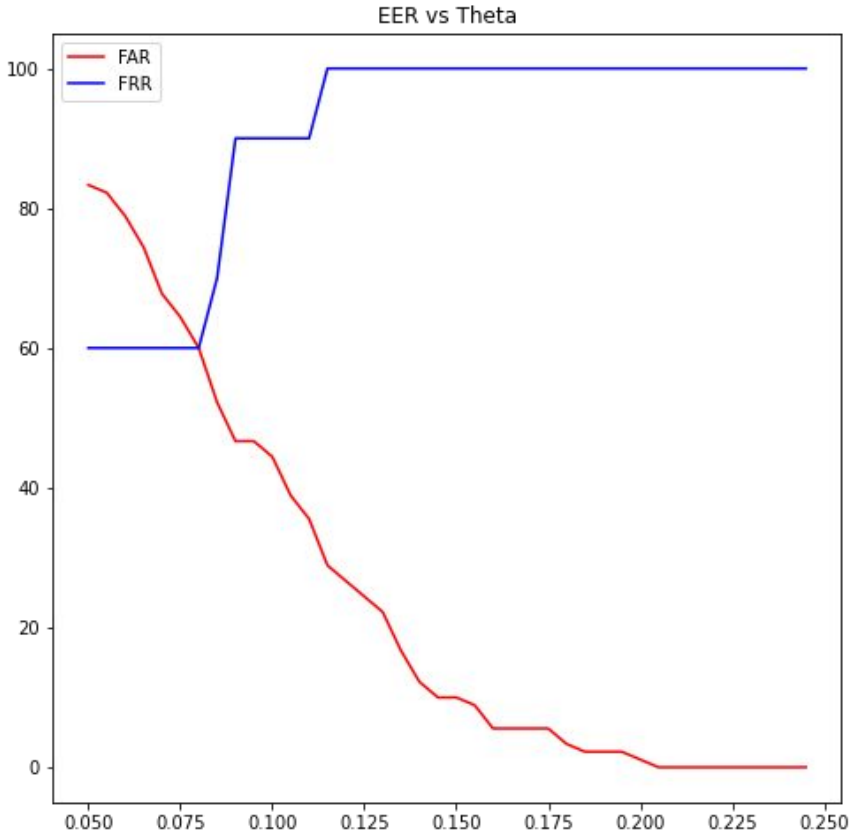
Train set for thresholding =0.10



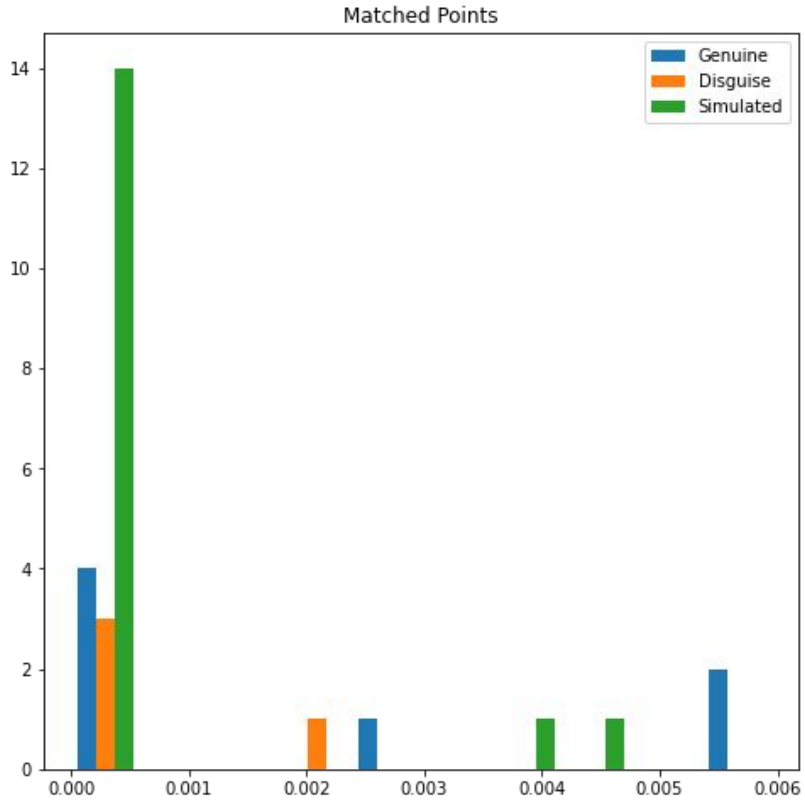


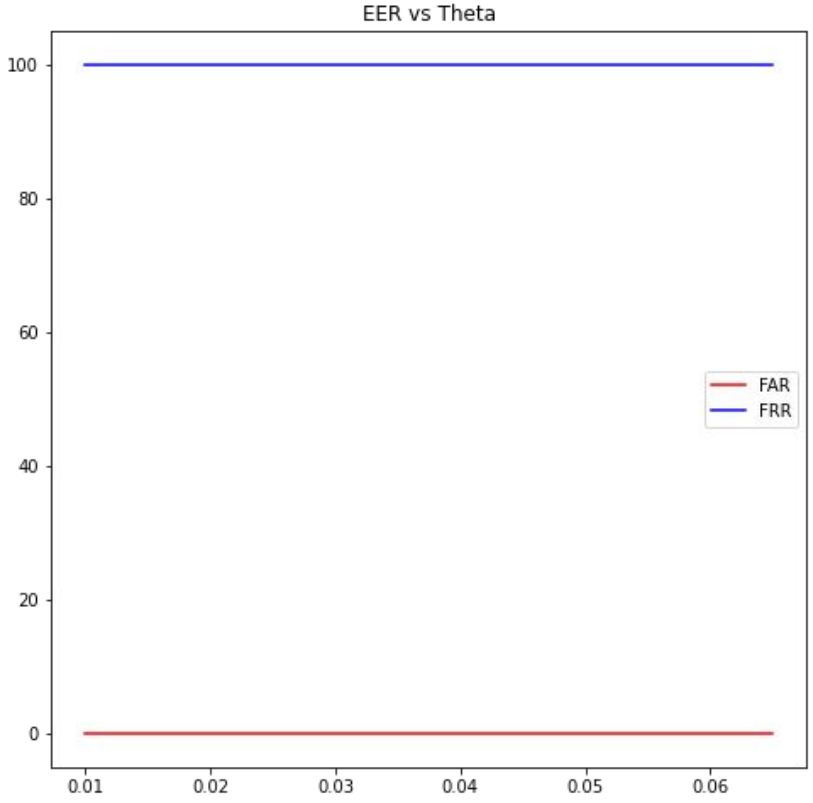
Test set for thresholding = 0.10



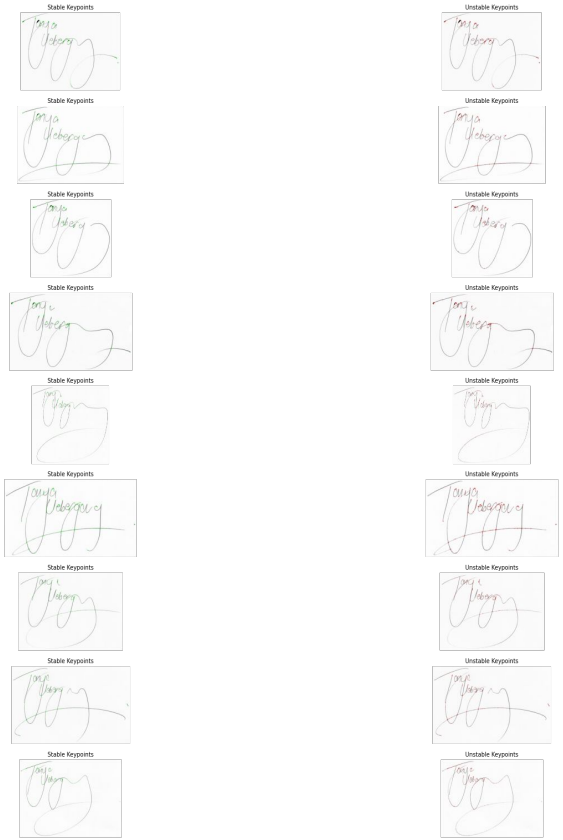


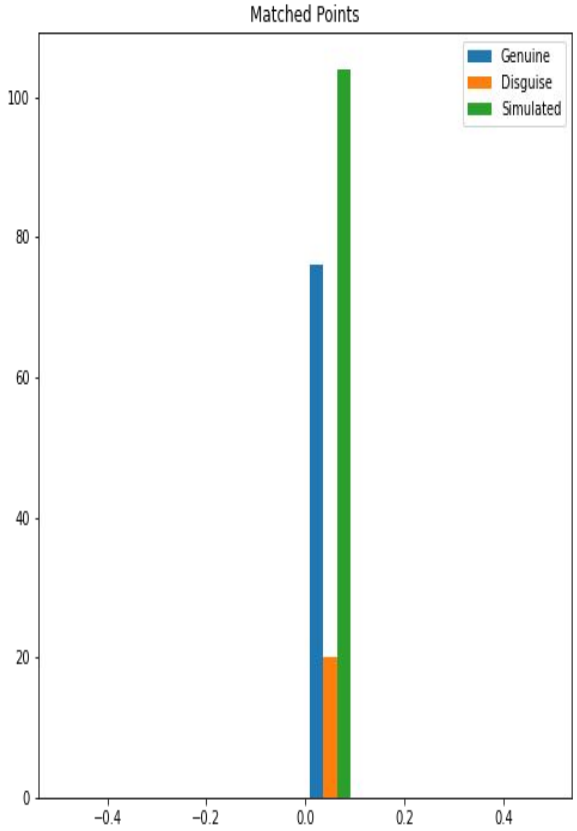
Generated set for threshold=0.10

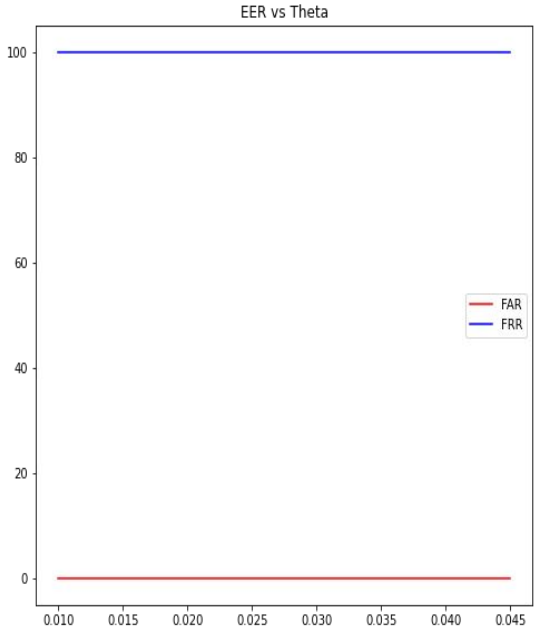




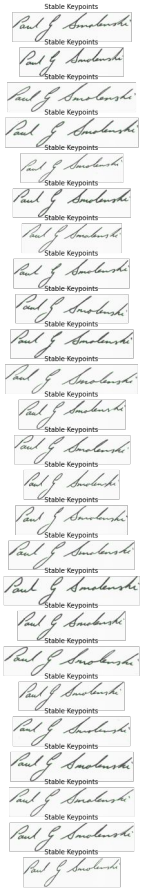
Train with orb :1500 features

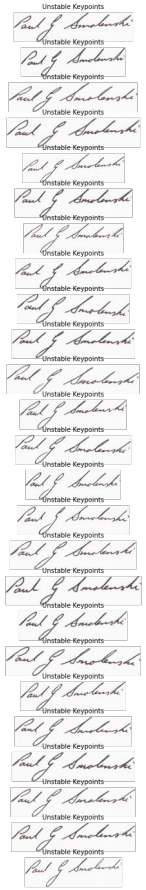


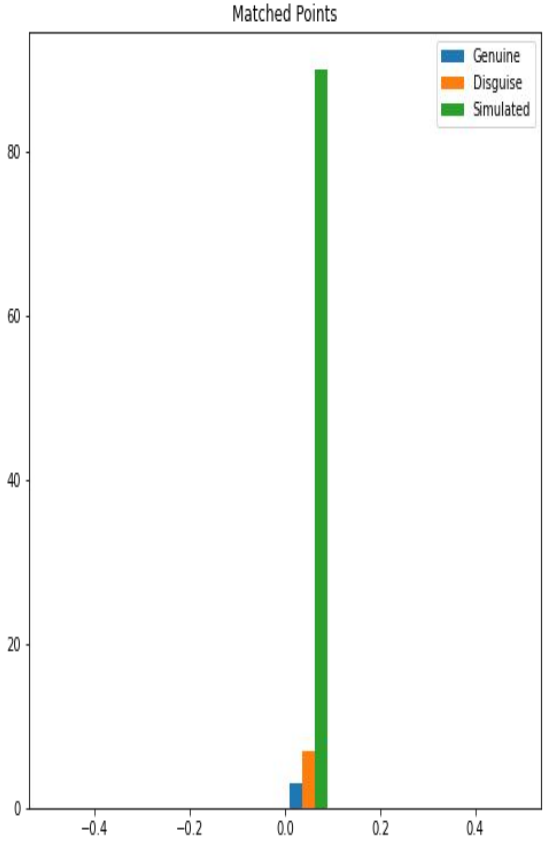


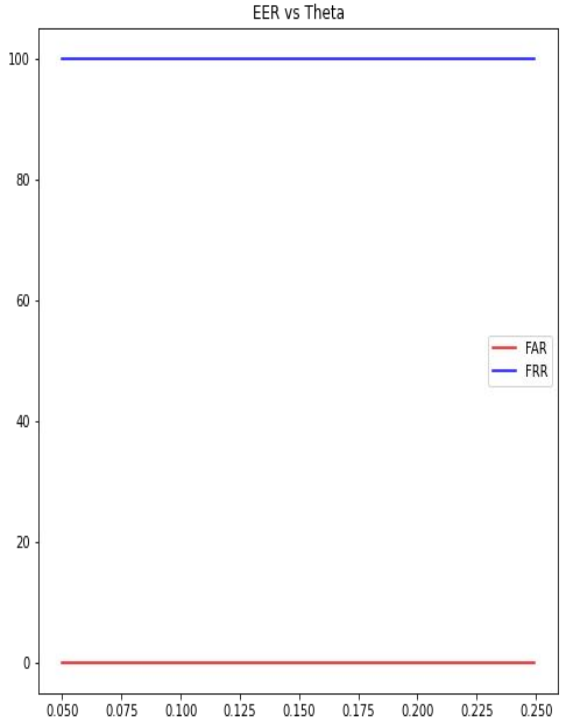


Test with orb :1500 features

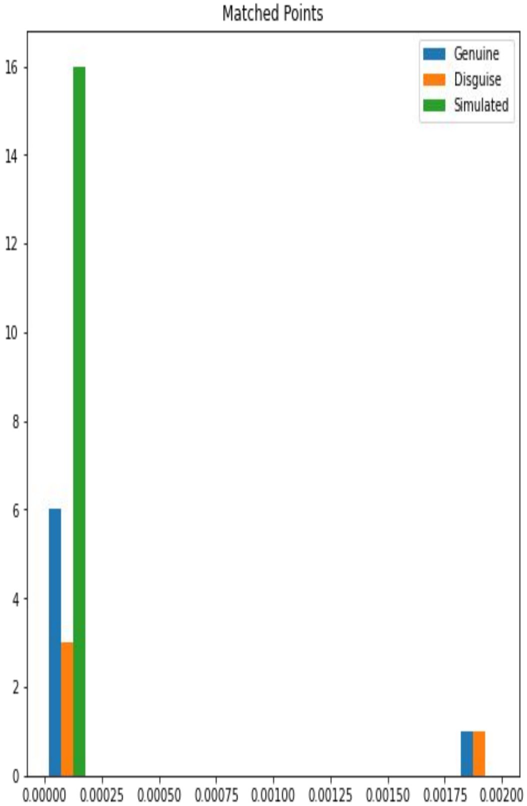


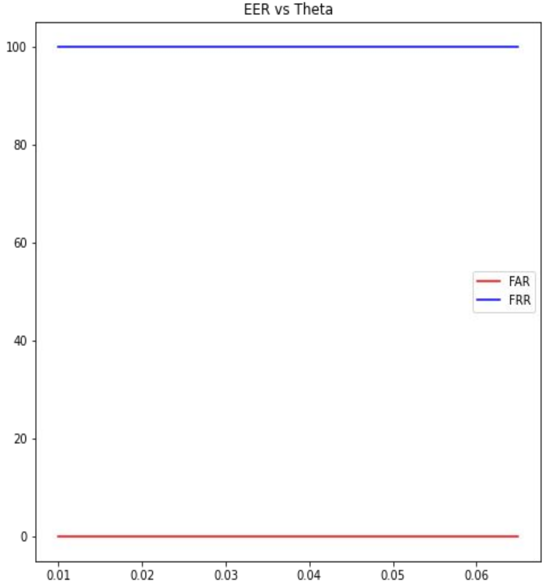






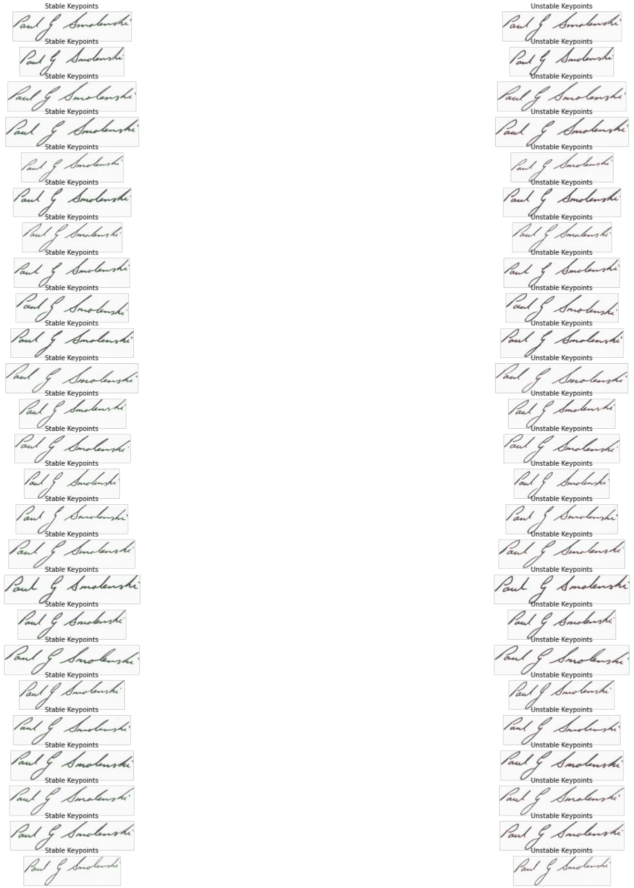
Generated set with orb : 1500 features





DB for sift :


Stable Keypoints
Unstable Keypoints


Stable Keypoints
Unstable Keypoints

Contribution

Feature Extraction(Database Creation) - Mohit

Classify and Validate - Suma

Training and Verification (own dataset)- Raja

Testing and Plots - Gowtami

Debugging and Documentation - Raja and Tejaswini

Presentation - Gowthami

THANK YOU

ANY QUESTIONS?