Biometrics System

Introduction

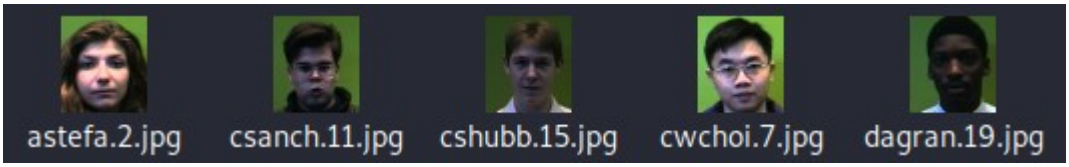
The goad of the project is to configure a biometrics system (face recognition) with the best configuration possible to a given data-set so that for future use and testing its completely ready to use.

Data-set

The data-set is taken from (https://cswww.essex.ac.uk/mv/allfaces/index.html) and a subset of images has been used for time reason.

The data-set is composed by a gallery and impostors in two separate folders and the genuine probes are obtained from the gallery, taking random images from different people (one per person).

Each persona in the data-set (gallery and impostors) has 20 images labeled with a name and and index separated with a '.' (dot).



[img1 - pattern of people name]

The impostors has the same patter of the name as the data-set, the only difference is that it has 'IMP' at the beginning of the file name to identify that that person is an impostor and needed for the ground truth.

[img2 – Impostor's name pattern]

There are two types of data-set for testing, the first one is well made where people has a set of similar images of themselves with little variation of face expression and a second data-set where images of people are very different and they vary in age, illumination a rotation.

The first data-set gallery is composed by ten female and ten male people on the other hand the impostors are composed by nine female and five male people.

All the people in the data-set has slightly different facial expression.

The second data-set is composed by all male people. 12 people in gallery and 12 impostors.

Functions implemented

CreateGallery

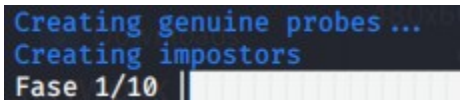
Paramenter: path

This function is created to dynamically populate the probes, depending on the path it takes, path of gallery or path of the impostors directory.

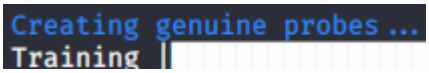
From the given path it take a certain amount of images (one per person), the amount of the number of images to take is initially set and hard coded.

The main utility of this function is to create dynamically a potential infinite number of testing probes by combining genuine probes from gallery and impostors.

With this function is possible to easily create dynamic open set and closed set.



[img3 – Creation of an open set]

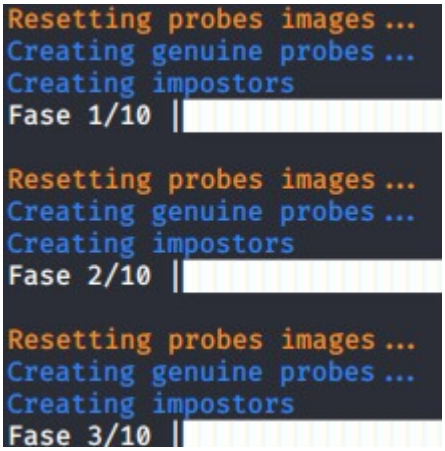


[img4 – Creation of an closed set]

ResetGallery

This function simply restore the images from the probes folder to the respective gallery or impostors folder.

Usually used before populating the probes while we want to maintain a constant and predefined number of probes, also to create other test probes.

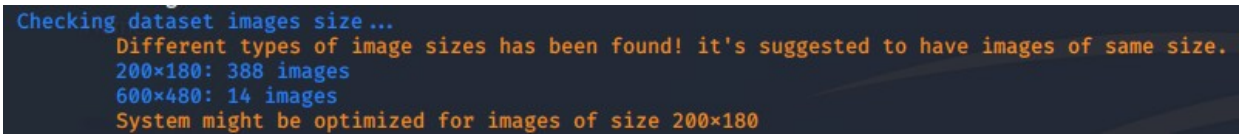


[img5 – Creation of open sets dynamically for testing]

DetectDatasetImgSize

This function is needed to detect if there are various type of image size.

While the system works better if all images in the data-set are the same size this function detect the most images with the same size and tries to adapt the system for them.



[img6 – Test with different size of image in gallery]

CalculateScaleFactor

This value specifying how much the image size is reduced at each image scale.

This scale factor is used to create scale pyramid as shown in the picture. Suppose, the scale factor is 1.03, it means we're using a small step for resizing, e.g. reduce size by 3%, it increase the chance of a matching size with the model for detection is found, while it's expensive.

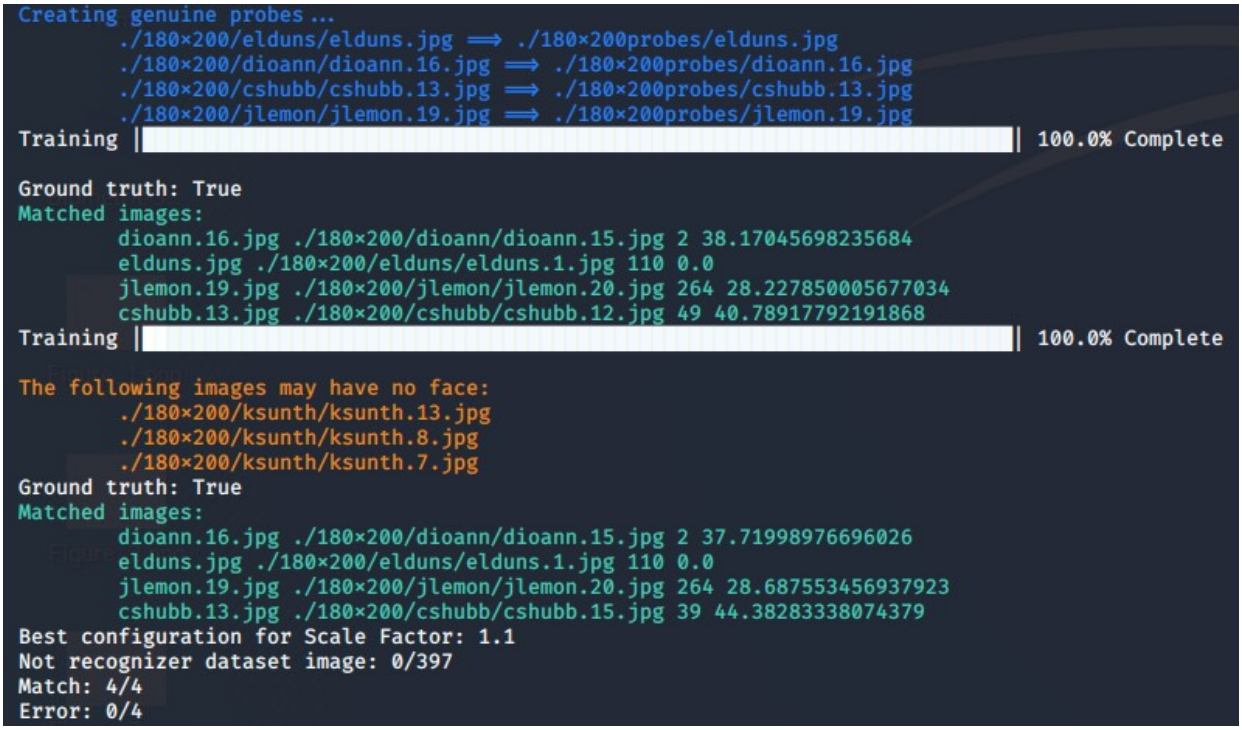
[img7 – Scale factor effect on images]

In this system the scale factor step is 0.1, the starting scale factor is 1.1 and at every try its increased by 0.1 until it find the best scale factor for which the system identifies the highest number of faces in the images and the most match of proves with the right gallery images.

In particular this function checks first of all of the number of matches is increased or decreased then it checks the number if identified faces in the mages and at last the number of non matched people.

If their values are positive at each step or equal to previews step the scale factor keep increasing in search for the best value otherwise it stops and the the value for the rest tests.

This function creates a new probe set at every step.

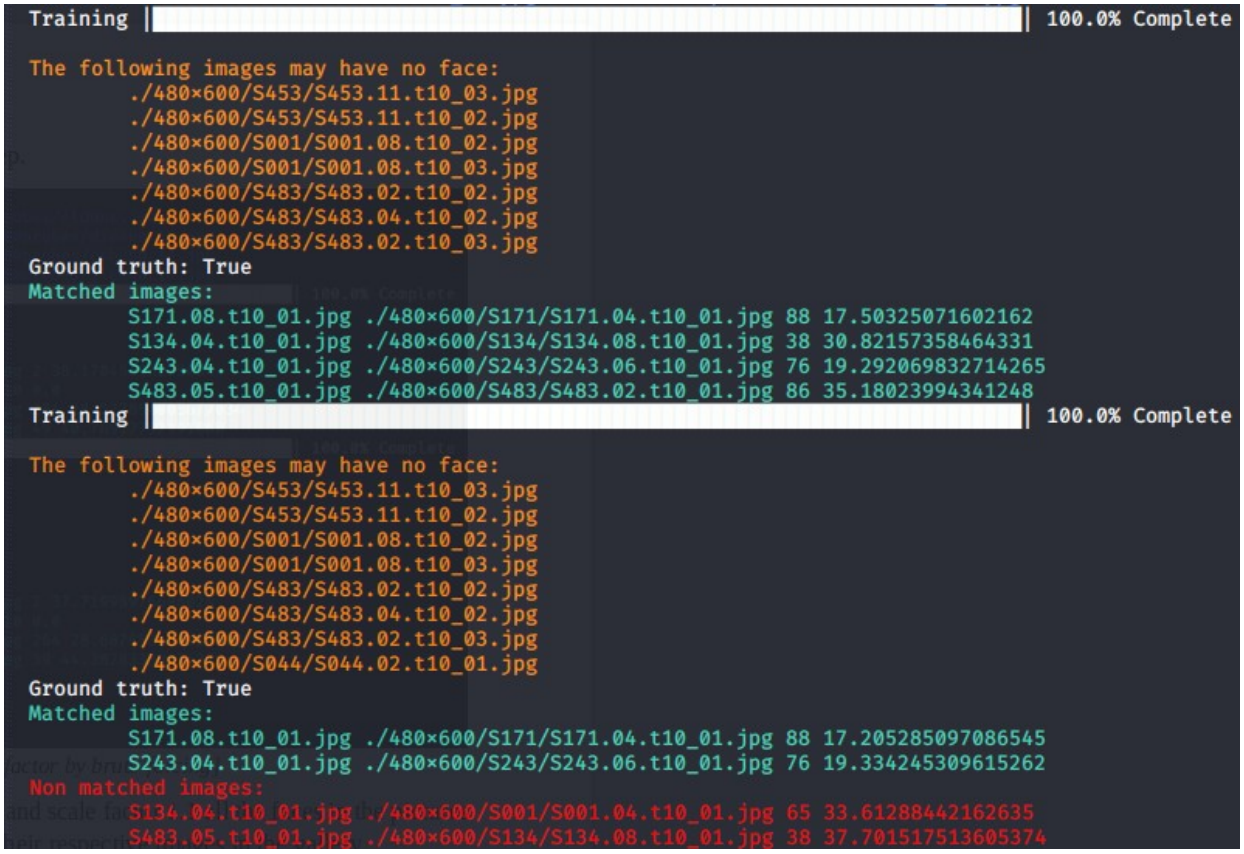


[img8 – Setting the scale factor by brute-forcing on the second data-set]

As the picture shows tested with the first data-set with probe number set to 4 and scale factor 1.1 all the faces in the pictures were identified and all the probes matched with their respective images in the gallery.

The check is done with the ground truth enabled so we know for sure if they match or not regardless of the set threshold.

At the second try with scale factor 1.2 in three images faces were not detected due to a high scale factor for the size of the images in the gallery (180x200) so the system take the first scale factor (1.1) and set it for the system.



[img9 – Setting the scale factor by brute-forcing on the second data-set]

As we see in this case with the second data-set at the first step with scale factor 1.1 there are 7 faces are not recognized form the gallery and all the probes are correctly matched.

But in the second step with scale factor 1.2, 8 faces are not recognized so worse than before but the matching become worse, in fact 2 probes match and 2 don't.

Now the way it behaves and the criteria it uses to calculate the scale factor can be changed depending on the need of the user.

SetThreshold

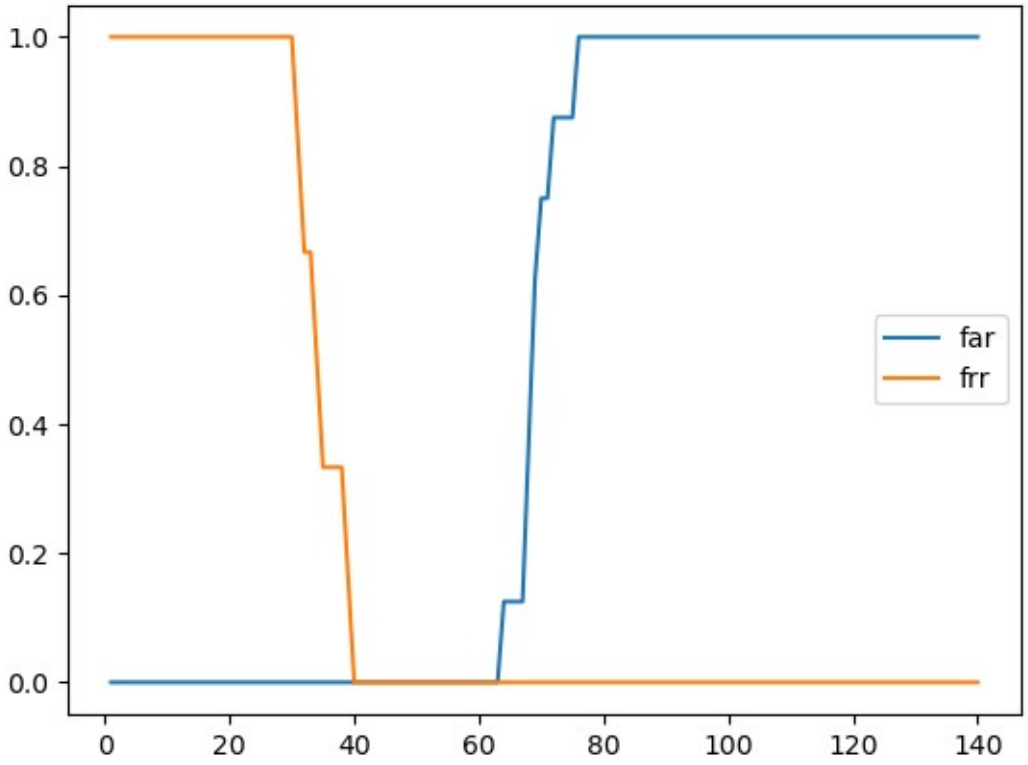
This function searches for the best threshold for the given data-set.

This function calculates FRR and FAR to set the right threshold with open set, the number of impostors and genuine are chosen randomly from 1 up to for a matter of speed otherwise it takes too long but these values can be changed any time..

The system calculates every thresholds starting from 140 to 0 and takes the value for which FRR and FAR are equal.

In case there are more than one value for which FRR is equal to FAR The system takes the average of these values.

At the end of this process the function draw the graph of the FRR and FAR.



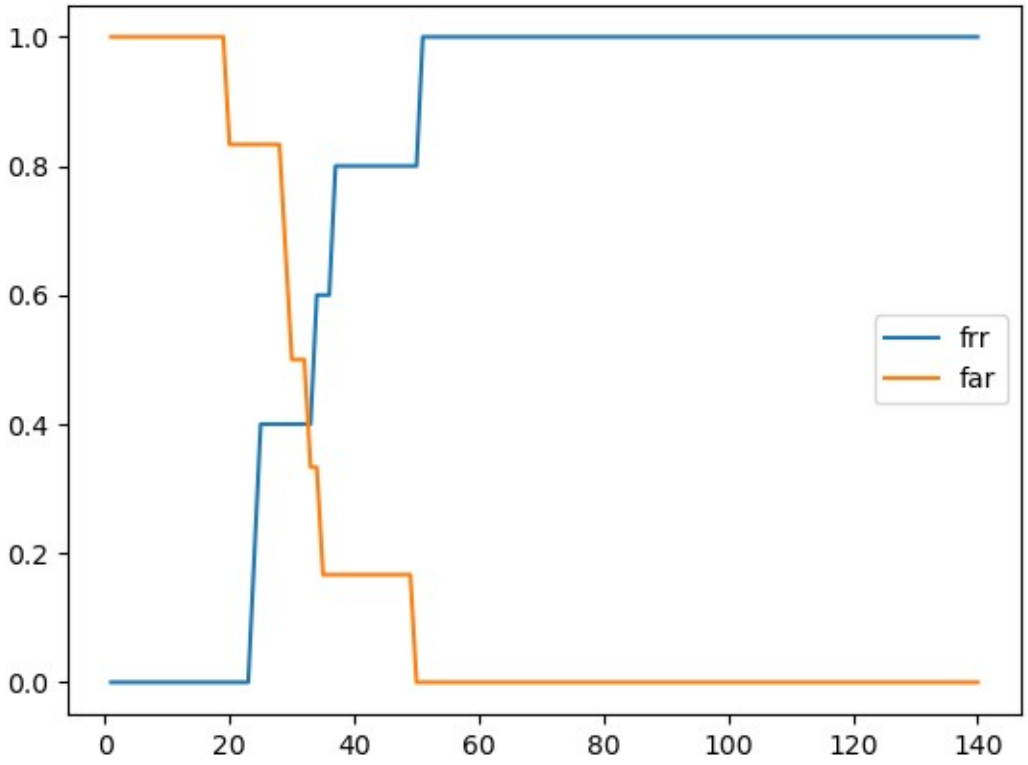
[img11 – FRR and FAR curve, threshold = 54]

Notice how the graph is not very curved and there are many value for which FRR and FAR are equal, in this case equal to 0.

This is caused because of the similarity between the images of each person.

In this case the threshold is the average of all the point where the FRR and FAR are equal.

With the following data-set the optimal threshold is 62 but a threshold of 52 is still acceptable because its better to have a little bit strict system than having false acceptance.



With the second data-set we have a different graph due to different type and quality of the images.

CMS (Cumulative Match Score)

This function calculated the probability and the ration of CMS to a given k or a range of k (from 1 to k).

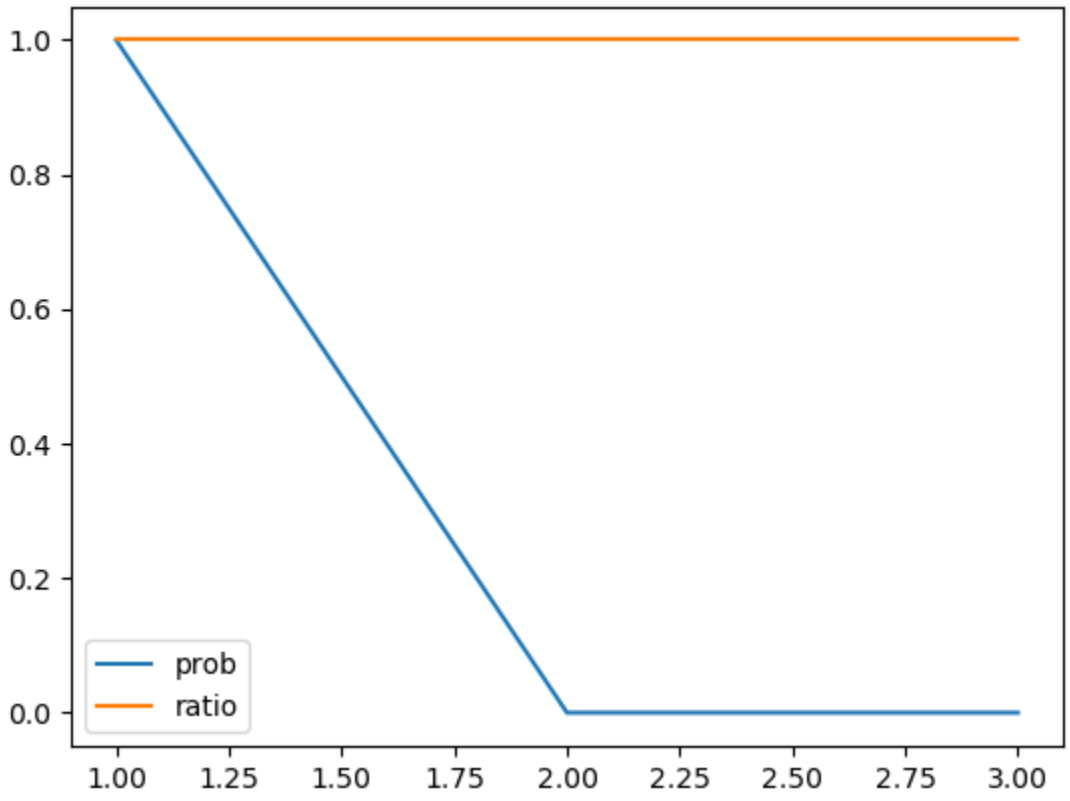
First of all we calculate the matrix (Probes X Gallery) and take the minimum distance of each person.

	A1	A2	A3	B1	B2	B3	C1	C2	C3		A1	B3	C3
Probe A	23.5	22.6	33.3	66.1	65.0	63.3	75.5	71.2	71.1		22.6	63.3	71.1

[img14 – One minimum distance per person is taken]

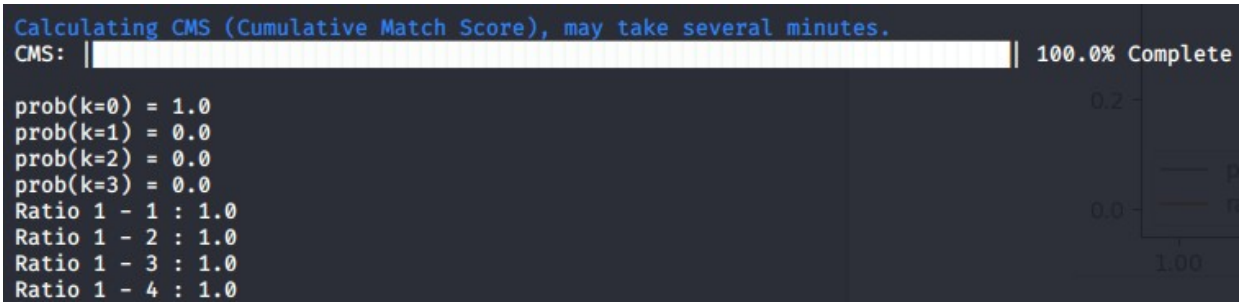
Once got the matrix we check the CMS from rank 1 to rank k and calculate it as #match at tank k / #probes at each rank.

The ratio is calculate for rank 1 to 1, 1 to 2… 1 to k and produce the graph with the first data-set.



[img15 – CMS ratio and prbability with k = 4]

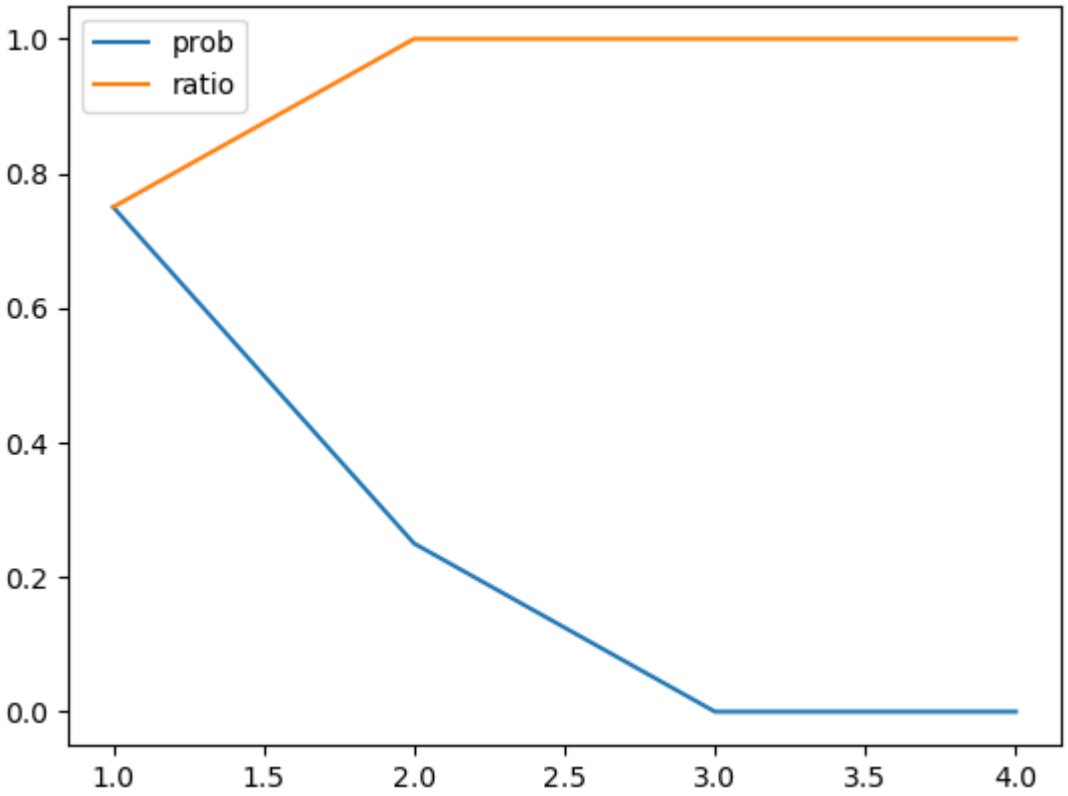
For the probability the is a 100% hit rate of the probes at rank 1 and 0% at rank 2 and 3, for the the ratio there is a 100% hit at every rank while it is the ratio between the number of individuals which are correctly recognized among the first k and the total number of individuals in the test set (probe).



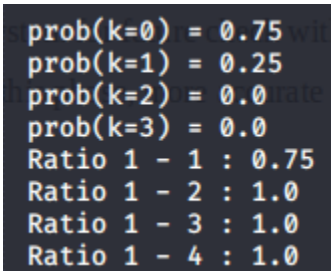
[img16 – Log of probability and ratio at each rank]

Here the img13 show the Log of the ratio and probability value at each rank.

Same test with the second data-set show a different graph.



[img17 – CMS ratio and probability with k = 4]

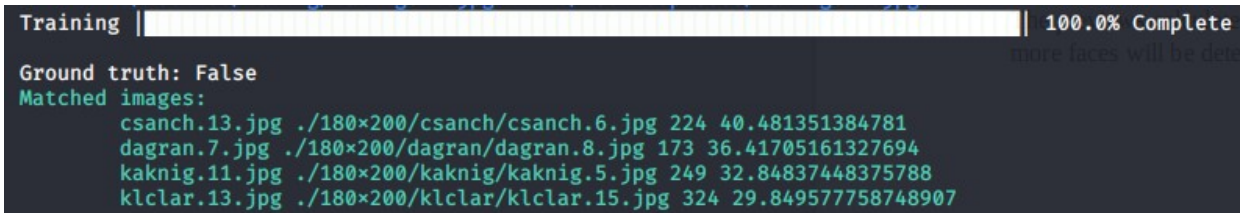


[img18 – Log of probability and ration at each rank]

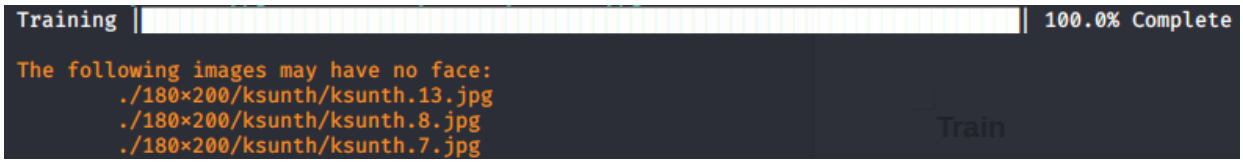
Train

This function enroll the gallery images into the system for future check with probes.

The previews calculated scale factor is crucial in this phase, more accurate is the scale factor more faces will be detected and enrolled.



[img19 – Successful training]



[img20 – Training with non optimal scale factor]

TopMatch

Function that return the best match (lowest distance) between probes and an image of the gallery.

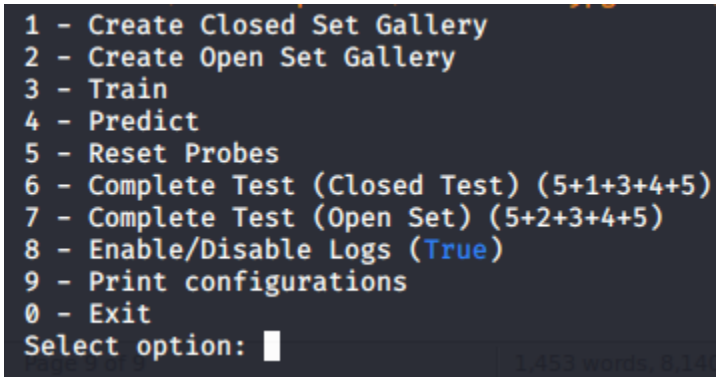
General Overview

The tool can be be launched with optional parameters:

-l : enable logs from the beginning (by default its disabled)

-i : interactive mode useful for presentation (stops at each step)

The system set all the parameters at the initial phase such as scale factor, threshold and CMS (Cumulative Match Score) than let the user to to make further test through a menu giving various types o options.



[img21 – Possible operations]

Once the initial phase set up all the values the used can make further test creating open set or closed set, creating as many different probe set as want and carry on further test to very the actual configuration with the help of the logs.

This tool can be the core of every biometrics system.

For example in case of development of a system that unlock a PC by taking the picture form the webcam, the picture can be sent to the tool already trained and then compare the result.

Or biometrics system for accessing room, also here the picture taken from the camera can be sent for identification or verification to the tool for the check.

The tool at the moment does only the identification but can be upgraded to do also the verification.