# Suitability Measure of Face Recognition System

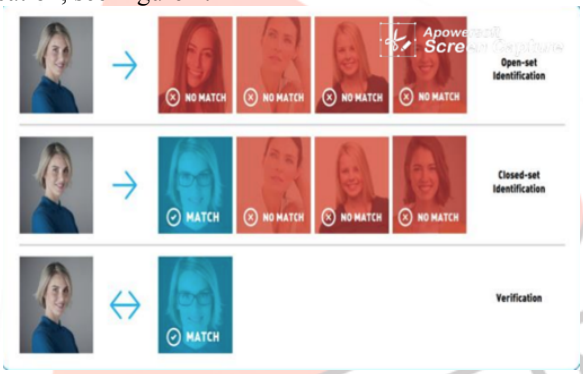
Anushtuthi Palani, Krishna Sridhar
Student, R.V. College of Engineering

**Abstract— In facial recognition systems, it is often seen that an appropriate measurement of system's accuracy is a difficult task. The lack of familiarity with the terminology and the testing algorithms is one the major reasons. A system suitable for one scenario may not work at all with another. It's not just the algorithm which is the decisive factor here but the raw data that we choose to train our model. In this paper we introduce the jargons and the suitability measure of raw data specific to the face recognition technology.**

**Keywords – FAR, FRR, ROC**

## I. INTRODUCTION

Facial recognition is referred to detecting a human face in a photograph or video and identifying it. It may also be termed for tasks such as determining sex and age, comparing two images and checking if they belong to the same person, etc. In this paper, we will focus on detection, identification and verification of a human faces. This is done by extracting special descriptors or feature vectors from the images. In this case, the identification problem is reduced to the search for the nearest vector of features, and verification can be realized with a simple threshold of distances between vectors. By combining these two actions, one can identify a person among a set of images or make a decision that he is not among these images. Such a procedure is called open-set identification, see Figure 1.




To quantify the similarity of individuals, you can use the distance in the space of characteristic vectors. Euclidean or cosine distance is often chosen, but there are other more complex approaches. A specific distance function is often supplied as part of the face recognition product. Identification and verification return different results and, accordingly, different metrics are used to assess their quality.

## II. ACCURACY VALUATION

*Dataset*

Dataset plays a major role in the overall build quality and effectiveness of a facial recognition system. As almost all of the facial recognition systems are a result of machine learning, the datasets decides the design and structure of the system. A natural way to verify that the accuracy of the face recognition algorithm meets expectations is to measure the accuracy on a separate test dataset. It is very important to choose this dataset correctly. Ideally, an organization should acquire its own set of data that is as much as possible similar to the images that the system will operate with. Pay attention to the camera, shooting conditions, age, sex and nationality of people who will get into the test data set. The more similar the test data for real data, the more reliable the test results will be. Therefore, it often makes sense to spend time and money to collect and mark up your data set. If this, for some reason, is not possible, you can use public datasets, for example, LFW and MegaFace. LFW contains only 6000 pairs of face images and is not suitable for many real scenarios: in particular, it is not possible to measure sufficiently low error rates on this dataset, as we will show below. Dataset MegaFace contains much more images and is suitable for testing face recognition algorithms on a large scale. However, both the training and test set of images of MegaFace is available for free, so use it for testing with caution. Retraining We will make a list of recommendations: do not use the data on which the algorithm was trained in testing, use a special closed dataset for testing. If this is not possible and you are going to use the public dataset, make sure that the vendor did not use it in the learning process and / or algorithm setup. Examine the data set before testing, consider how close it is to the data that will come from operating the system. Metrics After choosing a dataset, you need to determine the metric that will be used to evaluate the results. In general, a metric is a function that takes the results of an algorithm (identification or verification) on its input, and returns a number on the output that corresponds to the quality of the algorithm's performance on a particular dataset. Using a single number for quantitative comparison of different algorithms or vendors allows you to concisely represent the results of testing and facilitates the decision-making process. In this section, we'll look at the metrics most commonly used in face recognition, and discuss their meaning from a business

perspective. Verification Verification of persons can be considered as the process of making a binary decision: "yes" (two images belong to one person), "no" (a pair of photos depicts different people). Before dealing with verification metrics, it is useful to understand how we can classify errors in similar problems. Given that there are 2 possible algorithm responses and 2 versions of the true state of things, there are probably 4 possible outcomes:


In the table above, the columns correspond to the solution of the algorithm (blue - to accept, yellow - to reject), the lines correspond to the true values (coded by the same colors). The correct answers of the algorithm are marked with a green background, the wrong ones with a red one.

| | Accept | Reject |
|---|---|---|
| Reject | Type I error (False accept) | Correct (True reject) |
| Accept | Correct (True accept) | Type II error (False reject) |

Out of these outcomes, two correspond to the correct answers of the algorithm, and two correspond to the errors of the first and second kind, respectively. Errors of the first kind are called false accept, false positive or false match, and errors of the second kind are false reject, false negative or false nonmatch. Sum up the number of errors of all kinds among the pairs of images in the dataset and divide them by the number of pairs, we get false accept rate (FAR) and false reject rate (FRR). In the case of an access control system, "false positive" refers to granting access to a person for whom this access is not provided, while "false negative" means that the system erroneously denied access to an authorized person. These errors have different costs from the business point of view and are therefore considered separately. In the example with the "false negative" access control, the security officer needs to recheck the employee's pass. Providing unauthorized access to a potential offender (false positive) can lead to much worse consequences.

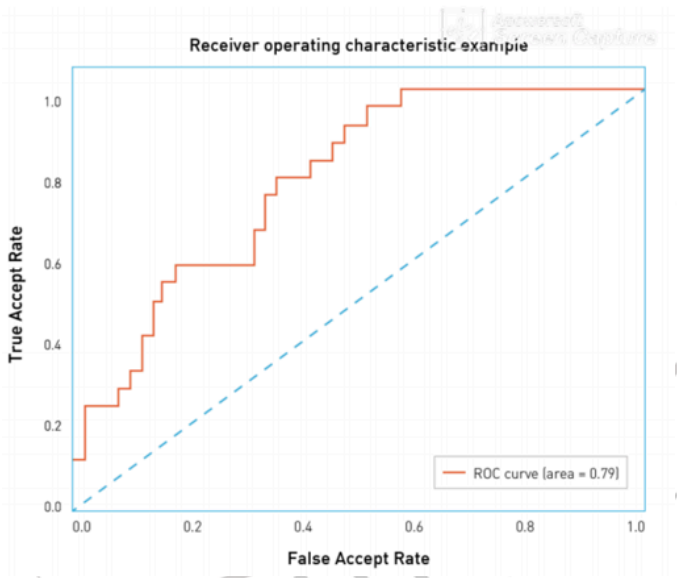
Given that errors of all kinds are associated with different risks, software manufacturers for face recognition often allow you to customize the algorithm so as to minimize one of the types of errors. For this, the algorithm returns not a binary value, but a real number that reflects the confidence of the algorithm in its solution. In this case, the user can independently select the threshold and fix the error level on certain values. For example, consider a "toy" dataset of three images. Let images 1 and 2 belong to the same person, and image 3 to someone else. Let's say that the program evaluated its confidence for each of the three pairs as follows:

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | | 0.85 | 0.6 |
| 2 | 0.85 | | 0.9 |
| 3 | 0.6 | 0.9 | |

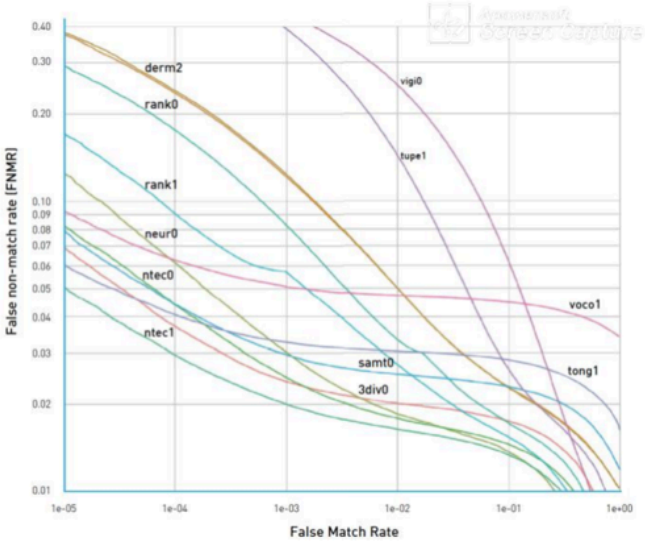
We specifically selected the values in such a way that no threshold would classify all three pairs correctly. In particular, any threshold below 0.6 will result in two false accept (for pairs 2-3 and 1-3). Of course, such a result can be improved. Selecting a threshold from the range from 0.6 to 0.85 will cause the pair 1-3 to be rejected, the pair 1-2 will still be accepted, and 2-3 will

be falsely accepted. If you increase the threshold to 0.85-0.9, then the 1-2 pair will be false rejected. Threshold values above 0.9 will result in two true reject (pairs 1-3 and 2-3) and one false reject (1-2). Thus, the best options are thresholds from the range 0.6-0.85 (one false accept 2-3) and a threshold above 0.9 (leads to false reject 1-2). What value to choose as the final depends on the cost of errors of different types. In this example, the threshold varies in a wide range, this is due, first of all, to very small dataset sizes and how we chose the confidence values of the algorithm. For large datasets used for real-world tasks, much more accurate threshold values would have been obtained. Often, face recognition software vendors supply default threshold values for different FARs that are computed in a similar fashion on the vendor's own data points. It is also not difficult to see that as the interesting FAR decreases, more and more positive pairs of images are required to accurately calculate the threshold value. So, for FAR = 0.001, at least 1000 pairs are needed, and for FAR = 0.0001, 1 million pairs is needed. It is not easy to assemble and mark such a data set, so customers interested in low FAR values should pay attention to public benchmarks, such as NIST Face Recognition Vendor Test or MegaFace. The latter should be treated with caution, since both the training and test samples are available to all comers, which can lead to an overly optimistic estimation of accuracy (see section "Retraining").

The types of errors differ in the cost associated with them, and the client has a way to shift the balance towards those or other errors. To do this, we need to consider a wide range of threshold values. A convenient way to visualize the accuracy of the algorithm for different FAR values is to construct ROC curves (English receiver operating characteristic). Let's look at how ROC curves are constructed and analyzed. The confidence of the algorithm (and hence the threshold) takes values from a fixed interval. In other words, these quantities are bounded above and below. Suppose this is an interval from 0 to 1. Now we can measure the number of errors by varying the threshold value from 0 to 1 with a small step. So, for each threshold value, we get the values FAR and TAR (true accept rate). Next, we will draw each point so that FAR corresponds to the abscissa axis, and TAR - to the ordinate axis.



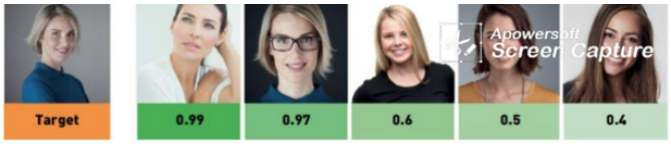

It is easy to see that the first point will have coordinates 1.1. With a threshold of 0, we accept all pairs and do not reject any. Similarly, the last point will be 0,0: at threshold 1 we do not accept any pair and reject all pairs. At other points the curve is usually convex. You can also notice that the worst curve lies approximately on the diagonal of the graph and corresponds to a random guessing of the outcome. On the other hand, the best possible curve forms a triangle with vertices (0,0) (0,1) and (1,1). But on a reasonable size, it's hard to come across.




It is possible to construct a similarity of RO-curves with different metrics / errors on the axis. Consider, for example, Figure 4. It shows that the organizers of the NIST FRVT on the Y axis have drawn FRR (False non-match rate in the figure), and on the X axis - FAR (in the figure - False match rate). In this particular case, the best results are achieved by curves that are below and shifted to the left, which corresponds to low FRR and FAR. Therefore, it is worth paying attention to what values are plotted along the axes. Such a graph makes it easy to judge the accuracy of the algorithm for a given FAR: it is sufficient to find a point on the curve with the X coordinate equal to the desired FAR and the corresponding value of TAR. The "quality" of the ROC curve can also be estimated in one number, for this it is necessary to calculate the area under it. The best possible value is 1, and the value 0.5 corresponds to random guessing. This number is called the ROC AUC (Area Under Curve). However, it should be noted that the ROC AUC implicitly assumes that the errors of the first and second kind are unambiguous, which is not always the case. If the price of errors differs, you should pay attention to the shape of the curve and those areas where FAR meets business requirements.

## III. IDENTIFICATION

The second popular task of face recognition is identification, or searching for a person among a set of images. The search results are sorted by the confidence of the algorithm, and the most likely matches fall into the top of the list. Depending on whether or not the person is present in the search base, the identification is divided into two subcategories: closed-set identification (it is known that the person is in the database) and open-set identification (the person you are looking for may not be in the database). Accuracy is a reliable and understandable metric for closed-set identification. In fact, accuracy measures the number of times a person was needed from the search results. How does this work in practice? Let's understand. Let's start with the formulation of business requirements. Let's say we have a web page that can host ten search results. We need to measure the number of times that the person sought is in the first ten replies of the algorithm. This number is called Top-N accuracy (in this particular case, N equals 10). For each trial, we determine the image of the person we are looking for, and the gallery in which we search, so that the gallery contains at least one more image of this person. We look through the first ten results of the search algorithm and check whether there is a person among them. To obtain accuracy, it is necessary to summarize all the trials in which the person sought was in the search results, and divide by the total number of tests.




Open-set identification consists of searching for people most similar to the desired image, and determining whether any of them are the desired person based on the confidence of the algorithm. Open-set identification can be considered as a combination of closed-set identification and verification, so on this task you can apply all the same metrics as in the verification task. It is also easy to see that open-set identification can be reduced to pairwise comparisons of the desired image with all images from the gallery. In practice, this is not used for reasons of computational speed. Face recognition software often comes with fast search algorithms that can be found among millions of people like milliseconds. The pairwise comparisons would take much longer.

## IV. PRACTICAL EXAMPLES

As an illustration, let's look at some common situations and approaches to testing face recognition algorithms.

***Retail store***

Let's say that a medium-sized retail store wants to improve its loyalty program or reduce the amount of thefts. It's funny, but from the point of view of face recognition it's about the same thing. The main objective of this project is to identify the

permanent buyer or an intruder from the camera as early as possible and transmit this information to the seller or security officer. Let the loyalty program cover 100 customers. This task can be considered as an example of open-set identification. Having estimated the costs, the marketing department came to the conclusion that an acceptable level of error is to accept one visitor per regular customer per day. If a store visits 1000 visitors a day, each of which must be verified with a list of 100 regular customers, then the necessary FAR will be 0.001. After determining the acceptable level of error, you should select the appropriate test data for testing. A good option would be to place the camera in a suitable place (vendors can help with a specific device and location). Comparing transactions of holders of cards of the constant buyer with images from the camera and carrying out manual filtration, store employees can assemble a set of positive pairs. It also makes sense to collect a set of images of random visitors (one image per person). The total number of images should roughly correspond to the number of visitors to the store per day. By combining both sets, you can get a dataset of both "positive" and "negative" pairs. To check the desired accuracy should be enough for about a thousand "positive" pairs. By combining different regular customers and casual visitors, you can collect about 100,000 "negative" pairs. The next step is to run (or ask the vendor to run) the software and gain the confidence of the algorithm for each pair from the data set. When this is done, you can build an ROC curve and make sure that the number of correctly identified regular customers meets business requirements.

***E-Gate at the airport***

Modern airports serve tens of millions of passengers a year, and the passport control procedure is daily held by about 300,000 people. Automation of this process will significantly reduce costs. On the other hand, it is extremely undesirable to miss the violator, and the airport administration wants to minimize the risk of such an event. FAR = 0.1 corresponds to ten offenders per year and seems reasonable in this situation. If the given FAR, FRR is 0.1, then the costs for manual document verification can be reduced by a factor of ten. However, in order to assess the accuracy at a given level of FAR, tens of millions of images will be needed. The collection of such a large data center requires considerable resources and may require additional coordination of the processing of personal data. As a result, investments in such a system can pay off too much. In this case, it makes sense to refer to the NIST Face Recognition Vendor Test test report, which contains a dataset with photos with visas. The airport administration should choose a vendor based on testing on this data center, taking into account the passenger traffic.

***Targeted mailing list***

So far, we have considered examples in which the customer was interested in low FARs, but this is not always the case. Imagine a camera equipped with an advertising stand in a large shopping center. The shopping center has its own loyalty program and would like to identify its participants who stopped at the stand. Then these customers could send targeted messages with discounts and interesting offers on the basis of what interested them on the stand. Let's say that the operation of such a system costs $ 10, while about 1000 visitors a day stop at the stand. The marketing department estimated the profit from each targeted email at $ 0.0105. We would like to identify as many regular customers as possible and do not disturb others too much. In order for this newsletter to pay off, the accuracy should be equal to the cost of the stand divided by the number of visitors and the expected income from each letter. For our example, the accuracy is. The shopping center administration could collect the data set in the manner described in the "Retail store" section and measure the accuracy as described in the "Identification" section. Based on the results of testing, you can decide whether it will be possible to extract the expected benefits with the help of a face recognition system.

***Video Support***

In this article, we mainly discussed the work with images and almost did not touch the streaming video. Video can be viewed as a sequence of static images, so metrics and approaches to testing accuracy on images are applicable to video. It is worth noting that the processing of streaming video is much more costly in terms of the calculations performed and imposes additional restrictions on all stages of face recognition. When working with video, you should conduct a separate performance test, so the details of this process are not addressed in this text.

***Common mistakes***

In this section, we would like to list common problems and errors that occur when testing face recognition software, and give recommendations on how to avoid them.

***Testing on insufficient data size***

You should always be careful when choosing a data source for testing face recognition algorithms. One of the most important properties of a dataset is its size. The size of the data needs to be selected based on business requirements and FAR / TAR values. "Toy" datasets from several images of people from your office will enable you to "play" with the algorithm, measure its performance or test non-standard situations, but on their basis it is impossible to draw conclusions about the accuracy of the algorithm. To test the accuracy, you should use reasonable size datasets.

***Testing with a single threshold value***

Sometimes people test the face recognition algorithm at one fixed threshold (often chosen by the manufacturer "by default") and take into account only one type of error. This is incorrect, because the default threshold values for different vendors differ or are selected based on different FAR or TAR values. When testing, you should pay attention to both types of errors.

***Comparison of results on different data sets***

Datasets differ in size, quality and complexity, so the results of the work of algorithms on different datasets cannot be compared. You can easily abandon the best solution just because it was tested on a more complex, than a competitor, dataset.

***Draw conclusions based on testing on a single dataset***

You should try to test on several sets of data. When choosing a single public dataset, you cannot be sure that it was not used when learning or adjusting the algorithm. In this case, the accuracy of the algorithm will be overestimated. Fortunately, the probability of this event can be reduced by comparing the results on different datasets.

## V. CONCLUSION

In this note, we described the main components of testing face recognition algorithms: data sets, tasks, corresponding metrics and common scenarios.

## VI. REFERENCES

[1] "Face Recognition Applications". Animetrics. Retrieved 2008-06-04.
[2] "Facial Recognition: Who's Tracking You in Public?". Consumer Reports. Retrieved 2016-04-05. [3] "Airport Facial Recognition Passenger Flow Management". hrsid.com.
[4] Bonsor, K. "How Facial Recognition Systems Work". Retrieved 2008-06-02.
[5] Smith, Kelly. "Face Recognition" (PDF). Retrieved 2008-06-04.
[6] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, 1993, (15)10:1042-1052
[7] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009 ([1] TM book)
[8] Williams, Mark. "Better Face-Recognition Software". Retrieved 2008-06-02.
[9] Crawford, Mark. "Facial recognition progress report". SPIE Newsroom. Retrieved 2011-10-06. [10] Kimmel, Ron. "Three-dimensional face recognition" (PDF). Retrieved 2005-01-01.