Face Identification by a Cascade of Rejection Classifiers

Quan Yuan, Ashwin Thangali, Stan Sclaroff
Computer Science, Boston University,
Boston, MA 02215,
{yq,tvashwin,sclaroff}@cs.bu.edu

Abstract

Nearest neighbor search is commonly employed in face recognition but it does not scale well to large dataset sizes. A strategy to combine rejection classifiers into a cascade for face identification is proposed in this paper. A rejection classifier for a pair of classes is defined to reject at least one of the classes with high confidence. These rejection classifiers are able to share discriminants in feature space and at the same time have high confidence in the rejection decision. In the face identification problem, it is possible that a pair of known individual faces are very dissimilar. It is very unlikely that both of them are close to an unknown face in the feature space. Hence, only one of them needs to be considered. Using a cascade structure of rejection classifiers, the scope of nearest neighbor search can be reduced significantly. Experiments on Face Recognition Grand Challenge (FRGC) version 1 data demonstrate that the proposed method achieves significant speed up and an accuracy comparable with the brute force Nearest Neighbor method. In addition, a graph cut based clustering technique is employed to demonstrate that the pairwise separability of these rejection classifiers is capable of semantic grouping.

1. Introduction

Face recognition is an important application for surveillance and access control. In [12], three primary face recognition tasks are listed:

- Verification: Am I who I claim to be?
- Identification: Who am I?
- Watch list: Are you looking for me?

The verification task is a binary classification problem, i.e. the answer is yes or no. The identification task is a multi-class classification problem, only one out of many possible hypotheses can possibly be correct. The watch list task can be thought of as an identification problem combined with a verification problem. The first step: to determine whether an individual resembles someone on the watch list is a multi-class problem, the second step: to verify that the match is correct is a binary problem.

Previous works [2, 11, 18] propose to use similarity measures to solve the recognition problem. For the verification task, with an acceptance threshold, the test input is compared with one or many image samples of the ID he claims to be. The similarity score is then compared with an acceptance threshold to decide yes or no. Intuitively, a good similarity measure should give similar face pairs high scores and dissimilar face pairs low scores. However, since the similarity measure is global for all pairs of faces, it needs to be robust to changes in extrinsic parameters like pose and illumination, and intrinsic parameters like expression, hair, glasses and other appearance changes. Obtaining such a similarity measure is a daunting task. Recently, Viola and Jones proposed a classification technique to distinguish similar from dissimilar faces [10]. For the identification task, nearest neighbor search using a similarity measure can be quite expensive for large datasets since each image in the dataset is compared to the test image to determine the best match.

The method proposed in this paper solves the identification task using rejection classifiers. A rejection classifier is a classifier capable of rejecting at least one class with high confidence. One-vs-all Fisher linear discriminants are employed to obtain such rejection classifiers. These classifiers are combined in a cascade. Each of the rejection classifiers can often reject multiple classes and hence the proposed cascade structure is very efficient. A Nearest Neighbor search is performed on the remaining set of persons to output the top $k$ ranked IDs most similar to a given input face.

In our experiments on the FRGC version 1 data, significant speedup of up to three and accuracy comparable to the brute force Nearest Neighbor search are demonstrated.

2. Related Work

Our method originates from [13] who proposed a directed acyclic graph (DAG) of pairwise classifiers for multi-class classification. In their approach, each pairwise classifier...
was a SVM. In an empirical study [9], the DAGSVM is shown to have very good accuracy when compared with more complex methods. In [5], DAGSVM is used for face recognition.

At each node of the DAG, a pairwise comparison is made and one of the classes is rejected. If the number of classes is \( M \), then the DAG needs \( M - 1 \) comparisons and hence is more efficient than methods like Pairwise Voting [3, 6], which needs \( \binom{M}{2} \) comparisons. Furthermore, the graph structure need not be stored, since the route that an instance takes can be determined during classification.

It is undesirable to train pairwise SVMs as in [5] for a large number of classes since the number of SVMs to be trained grows quadratically with the number of classes. For instance, in [5] 9316 SVMs were needed in an experiment with 137 classes. In contrast, we use one-vs-all Fisher linear discriminants [1] to obtain projections in the feature space. The number of projections learnt is linear in the number of classes. Each of the projections can often reject multiple classes and hence the proposed cascade structure is more efficient than the DAG which needs exactly \( M - 1 \) levels.

3. Cascade of Rejection Classifiers

In our framework each classifier rejects at least one class with high confidence. We call such classifiers rejection classifiers.

3.1. Rejection Classifiers

Given a projection \( w \), the threshold for a pair of classes \( A \) and \( B \) is selected to achieve a high margin between them on the training set. This ensures that a decision to reject a class at a node in the modified DAG is correct with high probability and hence the class need not be considered further. The margin between a pair of classes is defined by the separation between the classes in the projected space. Let \( x_0, x_1 \) be the set of training points for the two classes and \( p_0, p_1 \) be the projections of these points on \( w \). Let \( p_0 \) be to the left of \( p_1 \). The margin is given by

\[
\text{margin} = \frac{2(\min(p_1) - \max(p_0))}{\sigma_0 + \sigma_1}
\]

where \( \sigma_0, \sigma_1 \) are the standard deviations of \( p_0 \) and \( p_1 \). If the margin is greater than a predefined threshold, the pairwise classifier is retained. The threshold \( t \) to separate the classes is given by

\[
t = \frac{\min(p_1) - \max(p_0)}{2}.
\]

The rejection classifiers obtained in this way can be regarded as a subset of general pairwise classifiers.

3.2. Sharing Discriminants Among Classifiers

To reduce the number of classifiers that need to be trained, we observe that many pairwise classifiers can share the same linear discriminant. For instance, if person A can be separated from person B and C with sufficient margin by a Fisher discriminant, it is possible that the rejection classifiers (A,B), (A,C) and (D,B) can share the same discriminant, as shown in Figure 2.
possible projections. We instead use the Fisher linear discriminant obtained by solving the One-vs-All classification problem that maximizes the separation between a particular class (positive class) and the other classes (negative classes). Although on this projection the negative classes tend to overlap, they have good separations with the positive class.

Given the One-vs-All discriminant, each negative class is checked if a certain margin (Eqn. 1) is achieved with the positive class. If true, the rejection classifier for the negative class is given by a threshold along the discriminant (Eqn. 2). The threshold to reject the positive class is defined as the negative class threshold farthest from the positive class so as to maximize the confidence in this rejection. For some negative classes without sufficient margin, the rejection classifiers are not obtained. However, since there are \( M \) projections to be checked, it is possible that in other projections they are well separated.

Another advantage of using the One-vs-All discriminant to find a projection shared among many classifiers is the reduced chance of over-fitting and ill-conditioned matrices (during Fisher linear discriminant computation) that may result due to the small number of per class training samples.

4. Face Identification via a Cascade of Rejection Classifiers

In the identification problem, each person in the set of known IDs constitutes a class. Rejection classifiers are used in a cascade to reject with high confidence as many classes as possible. We are now left with a much smaller set of classes and can employ an expensive technique like Nearest Neighbor classification to obtain the best class. This is more efficient than brute force Nearest Neighbor search over the whole training set, since many classes were already rejected by the cascade.

For the nearest neighbor method, any similarity measure suitable for faces can be used. Zhao et. al [19] give a survey of many successfully used similarity measures. The focus of this paper is to demonstrate the significant speedup achieved by the cascade of rejection classifiers. We demonstrate this using Euclidean distance in the eigenspace as in [17] and show significant speedup over nearest neighbor classification without much loss in accuracy.

Researchers in the past [7, 14] have proposed indexing techniques to speedup nearest neighbor search. Such techniques do not scale well to high dimensions, with utility typically limited to less than 20 dimensions. We use a 150 dimensional feature space and hence such techniques are not applicable.

4.1. Training and Ordering Rejection Classifiers

As described in Section 3.2, for each class \( i \), a classifier \( C^i_0 \) that rejects \( i \), and a set of classifiers \( \{C^i_j\} \) that rejects a subset of the remaining classes is trained. All these classifiers share the same linear discriminant \( w \) but use different thresholds. Each such set of rejection classifiers \( \{C^i_0, C^i_j\} \) forms a stage in the cascade. The total number of cascade stages trained is at most \( M \) (the number of classes), since some of projections are not able to separate any class pair with sufficient margin. The stages are sorted in decreasing order of the number of classes they can reject (given by \( |\{C^i_0, C^i_j\}| \)). This ordering greedily rejects as many classes as possible. We refer to this sequence of classifiers as a Cascade of Rejection Classifiers (CRC).

The frequency of number of rejection classifiers in a stage of the cascade is shown in Figure 3. This distribution shows that in practice, a significant number classes are well separated from the other classes. This provides further justification that the cascade can use this separable nature of the classes to significantly narrow down the set of classes to concentrate on.

![Figure 3: Frequency of number of classes rejected with high margin at a stage of the CRC during training for the FRGC dataset.](image)

4.2. Classification of Test Inputs

There are two steps to classify a test input. The CRC is first applied to reduce the scope of possible classes to a small set. The Nearest Neighbor method is then used to determine
the best class. These steps are illustrated in Figure 4 and
described below.

**CRC classification** Given a test input \( x \), classifiers from the
cascade are considered in the sequence described above. At
each level of the cascade, the stage corresponding to an un-
rejected class \( i \) is chosen. The test input \( x \) is projected on
\( w_i, x^p = w_i^T x \) and the rejection classifiers \( \{C_i^0, C_i^1\} \) are
run. Each rejection classifier is just a comparison with a
threshold and hence is very fast.

Typically, we expect that some classes get rejected at
each iteration and the number of levels needed would be
smaller than \( M \), the number of classes (IDs). In the worst
case, none of the classes is rejected and \( M \) levels may be
needed. In our experiments with the FRGC version 1 data,
we found that the average the number of levels in the cas-
cade was \( < \frac{2M}{3} \). The set of possible classes are obtained at
the end of this step.

**NN classification** Given the set of possible classes, Near-
est Neighbor search is performed on training samples from
these classes to determine the best class.

![Diagram of classification process](image)

Figure 4: Classification of a test input using a Cascade of
Rejection Classifiers (CRC).

5. Experiments

The performance of the proposed technique is evaluated
with the FRGC version 1 dataset of still images. This
dataset contains 263 subjects with approximately 21 sam-
ple per person. The face detector [15]^1 was able to detect
5472 faces from a total of 5658 images. To accurately align
the faces, a PCA subspace of 50 dimensions learned from
300 correctly aligned faces is used to compute the recon-
struction error to find the best orientation, scale and trans-
lation for a detected face. The aligned faces are resized to
100 \( \times \) 100.

For face identification, we determined empirically that
150 dimensions with the largest eigenvalues in the PCA
subspace give the best accuracy for both Fisher discriminant
and nearest neighbor methods. The PCA space is learned
from a subset of 1600 randomly selected images, which al-
low the computation of eigenfaces in reasonable time.

The focus of our work is to evaluate the speedup over
nearest neighbor classification and we hence chose a simple
set of features obtained by projections on eigenfaces. More
sophisticated features [19] have been shown to achieve sig-
ificantly higher performance and it is part of our future
work to incorporate these features.

5.1. Experimental Setup

We compare the performance of the proposed technique
with brute force Nearest Neighbor search for different sizes
of the test set. These experiments demonstrate that the pro-
posed method gives significant speedup and achieves accu-
ragy close to the Nearest Neighbor method.

[^1]: http://vasc.ri.cmu.edu/NNFaceDetector/
Nearest Neighbor  Euclidean distance is used to rank all faces in the training set as in [17]. The ranked samples are mapped to their corresponding classes to obtain a ranked list of face IDs.

CRC  The choice of minimum acceptable margin (Eqn.1) strongly influences the tradeoff in accuracy versus speedup. Typically, increasing the margin leads to fewer errors in the rejection classifiers but also leads to a larger set of possible classes in the Nearest Neighbor classification step. We tried different margins and found that on the FRGC version 1 data set, a minimum margin of 3 gives a reasonable tradeoff. For CRC, the ranked output is obtained by nearest neighbor search on the set of classes remaining after running the cascade.

5.2. Identification Performance

The performance of the techniques and the speedup achieved by the CRC are shown in Figure 7. It is clear that the CRC reduces the scope of nearest neighbors search by up to 85% without significant loss in accuracy. A summary of statistics for the CRC is given in Table 1.

The errors in the CRC method may arise from two sources,  

- errors induced during rejection of classes by linear discriminants at each stage of the cascade, and  
- errors in Nearest Neighbor search on the subset of confused classes.

These errors are shown in Table 2. The fraction of errors due to the cascade increases with larger test set sizes since the confidence in rejection for the classifiers in the cascade decreases for smaller training sets.

6. Discovering Semantic Groups of Faces

Humans are able to distinguish male from female, young from old, Asian from European, yet the identity of the person is unknown. It seems to us that there maybe basic “classifiers” in the “feature space” of human perception that have semantic meaning. Since we have defined pairwise classifiers between subjects, we are interested to see if any semantic information is captured by these classifiers.

We consider each person as a node in a complete graph $G$. The edge weight for each pair of nodes is determined by their separation in the projected space and is given by,$$w_{ij} = \frac{\sigma_i + \sigma_j}{|m_i - m_j|}$$

where $\sigma_i, \sigma_j$ are the standard deviations and $m_i - m_j$ are the means of the two classes in the projected space. For each pair of classes there are two projections defined by their corresponding One-vs-All linear discriminants. The edge weight is given by the minimum of these weights since this gives the best separation.

Given $G$, we use the Normalized Graph Cuts [16] algorithm to separate the faces into clusters of roughly equivalent size. The clusters obtained are shown in Figure 8. When they are clustered into 4 groups, one group of female faces clearly stands out. Most of the Asian people appear in another group, yet it is not so obvious as the female group. It seems to us that the discriminant information between females and males is easily captured in the PCA feature space with linear discriminant.

Clustering classes can be useful to find better projections between groups of classes. It is an interesting future work to see how it can improve the separability between classes.

7. Conclusion and Future work

In this paper an efficient technique for combining rejection classifiers is proposed to speed up the NN search for face identification. A margin maintained in each rejection classifier ensures the accuracy during the rejection process. The cascade is able to narrow the search and hence achieves a significant speedup over NN search. The separability obtained with these classifiers is employed to find semantic meaning by spectral clustering.

In the experiments, performance was compared with brute-force NN retrieval. It should be noted that indexing structures like R-trees and K-d trees [7, 8, 14] that can be used to make nearest neighbor retrieval faster are inappropriate in high-dimensional features spaces, like that encountered in PCA-based face recognition. However, more recent techniques like locality sensitive hashing (LSH) [4] may faster NN retrieval, and experimental comparison of our approach and LSH-based NN retrieval will be conducted in future work. We also plan to investigate more sophisticated similarity measures and improved feature extraction methods.

References


| Test set size as fraction of dataset | 10% | 21% | 32% | 43% | 65% |
| CRC speedup | 1.58 | 1.58 | 1.60 | 2.38 | 3.35 |
| Fraction of training set samples for NN search | 39% | 34% | 29% | 25% | 14% |
| # of classes left for NN search | 72 | 63 | 35 | 45 | 25 |
| # of projections in CRC | 148 | 157 | 158 | 169 | 169 |
| CRC training time | 831s | 726s | 654s | 519s | 230s |
| CRC time/per test | 0.04s | 0.03s | 0.03s | 0.03s | 0.02s |

Table 1: Performance statistics of the proposed method (CRC) for different sizes of test set averaged over 5 trials. The second row gives the speedup achieved by the proposed algorithm over brute force NN. The third row gives the fraction of the number of samples in the training set over which NN search is performed. The fourth row gives the number of classes remaining after the rejection step. The fifth row gives the average number of linear projections performed in the CRC and corresponds to the average number of levels in the cascade. In the worst case CRC may need to perform 263 projections. The sixth and seventh row gives the training and test times.

| Test set size as fraction of dataset | 10% | 21% | 32% | 43% | 65% |
| CRC Cascade error | 30 (4.5%) | 82 (6.3%) | 125 (6.8%) | 225 (9.1%) | 694 (19.0%) |
| CRC NN error | 98 (14.9%) | 208 (16.0%) | 322 (17.5%) | 487 (19.6%) | 839 (23.0%) |
| CRC Total error (rank 1) | 19.4% | 22.3% | 24.3% | 28.7% | 42.0% |
| Brute force NN error (rank 1) | 19.6% | 22.3% | 24.9% | 29.1% | 41.9% |

Table 2: Errors at rank 1 induced by the Cascade and NN steps in the CRC method. The brute force NN search error is also shown for comparison.


Figure 7: Comparison of face identification accuracy for CRC and Nearest Neighbor methods with different sizes of test set on FRGC version 1 data. The y-axis represents the percentage of test examples with the correct class ranked within the corresponding value on the x-axis. **Speedup** gives the speedup in execution time of the CRC over Nearest Neighbor search on the test set. **#NN** gives the fraction of nearest neighbors evaluated by the CRC when compared to brute force NN search. The size of the whole dataset is 5472 with 263 subjects and 21 samples per subject. The results are averaged from 5 trials.


Figure 8: Clustering faces of people with Normalized Graph Cuts using pairwise separability. Cluster 1 captures mainly female subjects. Cluster 2 captures mainly asian subjects and cluster 3 captures mainly white subjects.