Learning to Recognize Faces in Realistic Conditions

Anonymous Author(s)

Affiliation

Address

e-mail

Abstract

In this work, we create a face dataset for evaluating face recognition techniques using a subset of the Labeled Faces in the Wild database. Our dataset contains 159 individuals with 10 images per person. The images contain a wide range of lighting, poses, and expressions. We use the building blocks of the Facerec framework to construct seven recognition models and evaluate their performance on the dataset. Three of the models evaluated have implementations in the OpenCV library. We find that the best recognition model uses local binary pattern histogram (LBP) features and a support vector machine classifier. This model obtains a 48.40% recognition success rate on the dataset. The best model available in OpenCV achieves 33.40% accuracy using LBP features and a euclidean nearest neighbor classifier.

1 Motivation

Face recognition is the task of identifying an individual from an image of their face and a database of known faces. Despite being a relatively easy task for most humans, unconstrained face recognition by machines remains an open and active area of research. The potential applications of this technology are wide ranging, including security, automated photo tagging, and identification of strangers as part of an augmented reality experience. Many recognition algorithms have been developed, but most only achieve high recognition results under controlled expression, pose, and lighting conditions. Some of these recognition algorithms are available in open source software packages. Unfortunately, there is not much information available on how these algorithms perform on real world datasets. This research aims to help guide individuals who don’t have significant knowledge of face recognition but who wish to apply the techniques. We evaluate seven different models created using the building blocks in the Facerec framework [1]. Three of the models evaluated are also available as part of the OpenCV library [2]. The experimental results provide a starting point for choosing the best model for a face recognition task.

2 Experimental dataset

To evaluate the performance of different facial recognition algorithms, a dataset of faces is derived from the Labeled Faces in the Wild (LFW) [3] database. LFW contains over 13000 images of 5749 unique individuals with identity labels. The photographs are collected from the internet using a Viola-Jones style face detector. The images in LFW have a wide variety of backgrounds, lighting conditions, face poses, and expressions. This makes the database more useful for evaluating real world performance of a recognition system than other databases, such as the AT&T Face Database [4], which only contains faces captured under controlled laboratory settings. Figure 1 displays the first five images of George W. Bush (top) and Bill Clinton from LFW.
The standard evaluation procedure presented by LFW investigates the task of face verification, or deciding whether two presented faces are from the same or different individuals. This research is more interested in the task of determining the identity of a face from an image given a list of possible identities. Therefore, we narrow down LFW to the subset of individuals (159) with 10 or more images. For individuals with more than 10 images, only the first 10 are used. To improve recognition performance, the images are taken from the LFW-a [5] version of the database, which preprocesses images by converting them to grayscale and rotating them using commercial face alignment software. In a final stage of preprocessing, we run OpenCV’s Viola-Jones style face detector on the images, crop down to the detected face, and normalize the image sizes to 100px * 100px. Figure 2 shows the three preprocessing stages.

Figure 2: The original LFW images George_W_Bush_0005.jpg and Bill_Clinton_0004.jpg are given on the left. The aligned images from LFW-a are shown in the middle. The final preprocessed images used in our experiments are shown on the right.

3 Techniques evaluated

Most facial recognition techniques consist of two stages. The first stage attempts to extract high level features from the raw pixels in the image. The second stage applies a classifier to the features to learn decision boundaries between individuals. This research investigates three different feature extraction methods: Eigenfaces, Fisherfaces, and Local Binary Pattern Histograms (LBP). On top of these features, we apply a nearest neighbor classifier with two different distance metrics: Euclidean and Cosine. Finally, we investigate applying a Support Vector Machine (SVM) classifier to the LBP features.
3.1 Eigenfaces

Eigenfaces [6] is a feature extraction technique that uses principle component analysis (PCA) to map face images from the high dimensional pixel intensity space to a lower dimensional face space. PCA selects projection directions to account for the greatest variation in the data, therefore minimizing reconstruction error. The principal components found by PCA can be visualized as face templates, or Eigenfaces. Feature vectors are constructed by representing the face images as a linear combination of these Eigenfaces. The first 5 Eigenfaces from the test dataset are shown in figure 3 below. The feature vectors can then be used for classification. The number of Eigenfaces to use for constructing the feature vectors is a free parameter of the technique.

![Figure 3: The first five Eigenface templates from the experimental dataset](image)

3.2 Fisherfaces

Fisherfaces [7] is similar to Eigenfaces in that it performs a linear mapping from the high dimensional image space to a lower dimensional subspace. The key difference is that Fisherfaces uses linear discriminant analysis (LDA) instead of PCA. LDA takes the class labels of the training images into account, and maps to a subspace that maximizes between class variance. The intuition here is that LDA preserves information that differentiates classes and discards information that accounts for variation within single classes. Similar to PCA, the technique generates a set of templates called Fisherfaces. Feature vectors consist of the contributions of each Fisherface to the reconstruction of the image. The first 5 Fisherfaces from the test dataset are shown in figure 4. Because the Fisherface templates focus on differences between individuals, it is harder to make out facial features in the templates. However, face outlines are visible upon close examination.

![Figure 4: The first five Fisherface templates from the experimental dataset](image)

3.3 Local Binary Pattern Histograms

LBP [8] focuses on local features unlike the global, template based techniques like Eigenfaces and Fisherfaces. With LBP, face images are first segmented into a spatial grid. Then, within each grid square, a 3 by 3 window of pixels is moved across the square. At each location, the intensity values of the outer pixels are compared with the center pixel. If the intensity is greater, the pixel is assigned the value 1. Otherwise, it is assigned the value 0. Collecting the assigned values of the 8 pixels gives an 8 bit binary code for the center pixel. This code contains local texture information such as the presence of an edge. A histogram of these codes is build for each grid square, and the histograms are concatenated to form the feature vector. The advantage of this technique is that it encodes low level texture information in the binary codes while preserving spatial information by using the grid. A visualization of the LBP features is provided in figure 5.
3.4 Nearest Neighbor Classifier

One of the most simple classifiers we can apply to perform recognition is a single nearest neighbor. When an unknown face image is presented to the model, it extracts a feature vector from the image, and then searches for the closest feature vector of a known face. For gauging distance between vectors, different metrics give different results. One option is to use euclidean distance, which is the square root of the sum of the squared component-wise differences in the vectors. Another option is to use the cosine similarity metric, defined as:

\[ \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \ ||\vec{b}||} \]

3.5 Support Vector Machine Classifier

The SVM is a popular machine learning classifier that learns class differentiating decision boundaries in the feature space. This research applies a multiclass SVM with a 2nd degree polynomial kernel function on top of the LBP features. The SVM has many free parameters, and selecting appropriate values for these parameters significantly impacts recognition performance [9]. More information on the tuning of the SVM is given in the next section. For our experiments, the thin wrapper on LIBSVM [9] provided by the Facerec framework was used.

4 Results

Our experiments evaluate seven different combinations of the features and classifiers described in the previous section. Several of the features and classifiers have free parameters that must be tuned to achieve optimal accuracy. Experiments showing the tuning of the models are presented below, followed by a comparison of the best configurations of the models.

4.1 Tuning hyperparameters for Eigenfaces and Fisherfaces

Both Eigenfaces and Fisherfaces have a free parameter in the number of components in each feature vector used to perform recognition. To find an optimal value for this parameter, we use two fold cross validation on the dataset while varying the number of components. A graph of the results is given in figure[3]. Based on these results, 160 is chosen as the number of components for comparing the techniques. While more components may slightly increase accuracy for some techniques, it also slows down classification performance.
Applying an SVM classifier is investigated for the LBP feature extraction technique. A grid search is performed over part of the parameter space, and the two fold cross validation recognition accuracy results are shown in figure 7. The SVM chosen for the final evaluation uses a second degree polynomial kernel with regularization parameters $C = 8.0$ and $\gamma = 0.5$.

Using the optimal values of the parameters discovered above, we compare the recognition performance of the models using five fold cross validation. For each fold, 8 images from each of the 159
individuals are used to train the model. The remaining 2 images per individual are presented to the model for recognition. We report the mean recognition accuracy and the standard deviation from the five fold cross validation trials. The results are sorted in order of ascending accuracy in table 1 below.

Table 1: Recognition Performance Across the Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Recognition Accuracy</th>
<th>Available in OpenCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guessing</td>
<td>0.63%</td>
<td>Yes</td>
</tr>
<tr>
<td>Fisherfaces with Euclidean Distance</td>
<td>8.21% ± 0.12%</td>
<td>Yes</td>
</tr>
<tr>
<td>Eigenfaces with Euclidean Distance</td>
<td>13.70% ± 0.54%</td>
<td>Yes</td>
</tr>
<tr>
<td>Eigenfaces with Cosine Distance</td>
<td>16.82% ± 1.36%</td>
<td>No</td>
</tr>
<tr>
<td>Fisherfaces with Cosine Distance</td>
<td>27.12% ± 1.38%</td>
<td>No</td>
</tr>
<tr>
<td>LBP with Euclidean Distance</td>
<td>33.40% ± 1.28%</td>
<td>Yes</td>
</tr>
<tr>
<td>LBP with Cosine Distance</td>
<td>40.88% ± 0.82%</td>
<td>No</td>
</tr>
<tr>
<td>LBP with SVM Classifier</td>
<td>48.40% ± 2.29%</td>
<td>No</td>
</tr>
</tbody>
</table>

5 Discussion and conclusion

From the experiments performed above, it is clear that the LBP feature outperformed Eigenface and Fisherface features, regardless of which of the classifiers was used. The type of classifier used also had significant impact on the performance of the models. In all cases that used a nearest neighbor classifier, cosine similarity was a better distance metric than euclidean distance. Using an SVM classifier with the LBP features provided the highest recognition accuracy at 48.40%.

The results suggest LBP with euclidean nearest neighbor as the best model for individuals attempting face recognition using OpenCV. This model achieves 33.40% accuracy. The results can be improved by using a more complex classifier, and this may be a good way for OpenCV to boost the accuracy in future releases of the library.

A recognition model with an accuracy of 48.40% on a dataset of 159 individuals could be useful in applications, especially if the interface provides multiple identity suggestions. It would be very slow and challenging for a human to learn new identities of 159 individuals from photos to the point where they could achieve high recognition accuracy. It is also interesting to note that the best model is approximately 77 times more accurate than random guesses. However, the results demonstrate that there is still a lot of room for improvement in face recognition techniques.

References