Face Recognition Attendance System

V Shashank Bharadwaj , Parth Badgujar, Aryan Dushing[a]

[a] *Indian Institute of Terchnology, Roorkee, , Uttarakhand,*

Abstract

An introduction to our project, and a basic literature review of facial recognition.

1. Introduction

The objective is simple: To capture a picture of a student's face, and update their status for the occurring class to 'present'. This vastly reduces the time to be spent by a professor on actually noting the attendance down.

Attendance is quite a hassle to handle if it has to be handled manually. Thus the need for systems that do these jobs for us. This increases efficiency of production and work efficiency as both employee and employer have to focus less on the mundane task.

The steps involved in this process are simple: Capture the image. Take the face present in the image (face detection). Recognize the identity of the person (face identification) Update their status to 'present' in a database.

The task of face detection plus face identification together is face recognition.

2. Face Recognition - a Literature Review

2.1. More Introduction

At present, one needs a PIN to get cash from an ATM, a password for a computer, a dozen others to access the internet, and so on. Although very reliable methods of bio-metric personal identification exist, for example, fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant's cooperation or knowledge. As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research.

A general statement of the problem of machine recognition of faces can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Collateral information such as race, age, gender, facial expression, or speech may be

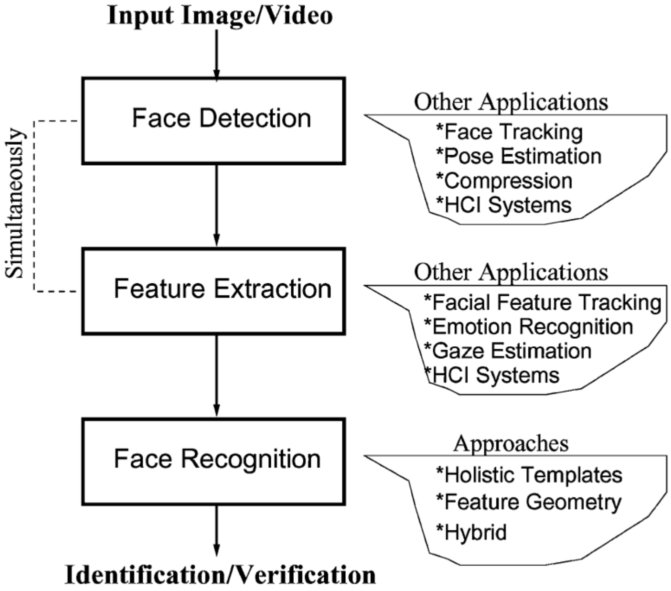

Fig. 1. Configuration of a generic face recognition system.

used in narrowing the search (enhancing recognition). The solution to the problem involves segmentation of faces (face detection) from cluttered scenes, feature extraction from the face regions, recognition, or verification.In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

During the past 8 years, face recognition has received increased attention and has advanced technically. Many commercial systems are now available to perform the same.

Here, we have written in detail about image identification from still images, and not videos.

2.2. The basic steps involved:

The problem of automatic face recognition involves three key steps/ sub-tasks: (1) detection and rough normalization of faces (2) feature extraction and accurate normalization of faces (3) identification and/or verification. Sometimes, different sub-tasks are not totally separated. For example, the facial features(eyes, nose, mouth) used for face recognition are often used in face detection as well. Face detection and feature extraction can be achieved simultaneously.

Research is done on each individual sub-task as well, as some of them are critical in certain applications.

The most noteworthy thing is that, given enough training images and time, any model made to recognize faces will accomplish its task successfully. Thus the actual issue we deal with is to get models that give accurate results even with very limited training images, even if the faces to be identified are in a different orientation than in the training models.

2.3. Face detection and face extraction:

2.3.1. Face Detection

Here, we only provide a summary and highlight a few recent models that are currently used and elicit good results.After a face has been detected, the task of feature extraction is to obtain features that are fed into a face classification system. Depending on the type of classification system, features can be local features such as lines or fiducial points, or facial features such as eyes, nose, and mouth. Face detection may also employ features, in which case features are extracted simultaneously with face detection. Feature extraction is also a key to animation and recognition of facial expressions.

Up to the mid-1990s, most work on segmentation was focused on single-face segmentation from a simple or complex background. These approaches included using a whole-face template, a deformable feature-based template, skin color, and a neural network.

Significant advances have been made in recent years in achieving automatic face detection under various conditions. Compared to feature-based methods and template-matching methods, appearance or image-based methods [Rowley et al. 1998; Sung and Poggio 1997] that train machine systems on large numbers of samples have achieved the best results. This may not be surprising since face objects are complicated, very similar to each other, and different from non-face objects. Through extensive training, computers can be quite good at detecting faces.

More recently, detection of faces under rotation in depth has been studied. One approach is based on training on multiple view samples [Gu et al. 2001; Schneiderman and Kanade 2000]. Compared to invariant-feature-based methods [Wiskott et al. 1997], multiview-based methods of face detection and recognition seem to be able to achieve better results when the angle of out-of-plane rotation is large (35○). In the psychology community, a similar debate exists on whether face recognition is viewpoint-invariant or not. Studies in both disciplines seem to support the idea that for small angles, face perception is view-independent, while for large angles, it is view-dependent.

In a detection problem, two statistics are important: true positives (also referred to as detection rate) and false positives (reported detections in non-face regions). An ideal system would have very high true positive and very low false positive rates. In practice, these two requirements are conflicting. Treating face detection as a two-class classification problem helps to reduce false positives dramatically [Rowley et al. 1998; Sung and Poggio 1997] while maintaining true positives. This is achieved by retraining systems with false positive samples that are generated by previously trained systems.

2.3.2. Feature Extraction

Three types of feature extraction methods can be distinguished: (1) generic methods based on edges, lines, and curves; (2) feature-template-based methods that are used to detect facial features such as eyes; (3) structural matching methods that take into consideration geometrical constraints on the features. Let us skip ahead directly to face recognition techniques.

3. Techniques of face recognition

3.1. Eigenfaces

Eigenface is one of the most thoroughly investigated approaches to face recognition. It is also known as Karhunen-Loève expansion, eigenpicture, eigenvector, and principal component. They argued that any face images could be approximately reconstructed by a small collection of weights for each face and a standard face picture (eigenpicture). The weights describing each face are obtained by projecting the face image onto the eigenpicture.

In mathematical terms, eigenfaces are the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. The eigenvectors are ordered to represent different amounts of the variation, respectively, among the faces. Each face can be represented exactly by a linear combination of the eigenfaces. It can also be approximated using only the "best" eigenvectors with the largest eigenvalues. The best M eigenfaces construct an M dimensional space, i.e., the "face space". The authors reported 96 percent, 85 percent, and 64 percent correct classifications averaged over lighting, orientation, and size variations, respectively. Their database contained 2,500 images of 16 individuals.

3.2. *Neural Networks*

This is the solution we have opted for in our model for the attendance system.

The attractiveness of using neural networks could be due to its non linearity in the network. Hence, the feature extraction step may be more efficient than the linear Karhunen-Loève methods. One of the first artificial neural networks (ANN) techniques used for face recognition is a single layer adaptive network called WISARD which contains a separate network for each stored individual. The way in constructing a neural network structure is crucial for successful recognition. It is very much dependent on the intended application. For face detection, multilayer perceptron and convolutional neural network have been applied. For face verification, multi-resolution pyramid structure. The SOM(self organizing map) provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimension reduction and invariance to minor changes in the image sample. The convolutional network extracts successively larger features in a hierarchical set of layers and provides partial invariance to translation, rotation, scale, and deformation. The authors reported 96.2 percent correct recognition on ORL database of 400 images of 40 individuals. The classification time is less than 0.5 second, but the training time is as long as 4 hours.

The PDBNN can be applied effectively to 1) face detector: which finds the location of a human face in a cluttered image, 2) eye localizer: which determines the positions of both eyes in order to generate meaningful feature vectors, and 3) face recognizer. PDNN does not have a fully connected network topology. Instead, it divides the network into K subnets. Each subset is dedicated to recognize one person in the database. PDNN uses the Gaussian activation function for its neurons, and the output of each "face subnet" is the weighted summation of the neuron outputs. In other words, the face subnet estimates the likelihood density using the popular mixture-of-Guassian model. Compared to the AWGN scheme, mixture of Guassian provides a much more flexible and complex model for approximating the time likelihood densities in the face space. The learning scheme of the PDNN consists of two phases, in the first phase; each subnet is trained by its own face images. In the second phase, called the decision-based learning, the subnet parameters may be trained by some particular samples from other face classes. The decision-based learning scheme does not use all the training samples for the training. Only misclassified patterns are used. If the sample is misclassified to the wrong subnet, the rightful subnet will tune its parameters so that its decision-region can be moved closer to the misclassified sample. PDBNN-based biometric identification system has the merits of both neural networks and statistical approaches, and its distributed computing principle is relatively easy to implement on parallel computer. It has been reported that PDBNN face recognizer had the capability of recognizing up to 200 people and could achieve up to 96 percent correct recognition rate in approximately 1 second. However, when the number of persons increases, the computing expense will become more demanding. In general, neural network approaches encounter problems when the number of classes (i.e., individuals) increases. Moreover, they are not suitable for a single model image recognition test because multiple model images per person are necessary in order for training the systems to "optimal" parameter setting.

3.2.1. *Graph matching*

Graph matching is another approach to face recognition. Dynamic link architecture is an extension to classical artificial neural networks. Memorized objects are represented by sparse graphs, whose vertices are labeled with a multi resolution description in terms of a local power spectrum and whose edges are labeled with geometrical distance vectors. Object recognition can be formulated as elastic graph matching which is performed by stochastic optimization of a matching cost function. They reported good results on a database of 87 people and a small set of office items comprising different expressions with a rotation of 15 degrees. The matching process is computationally expensive, taking about 25 seconds to compare with 87 stored objects on a parallel machine with 23 transputers. Encouraging results on faces with large rotation angles were obtained. They reported recognition rates of 86.5percent and 66.4percent for the matching tests of 111 faces of 15 degree rotation and 110 faces of 30 degree rotation to a gallery of 112 neutral frontal views. In general, dynamic link architecture is superior to other face recognition techniques in terms of rotation invariance; however, the matching process is computationally expensive.

Other than these , Hidden Markov Models(HMMs) ,Geometrical Feature Matching, Template Matching, 3D Morphable Model are also techniques that have been developed in the search for a perfect recognition system.

The following ones are more recent techniques

3.2.2. *Line Edge Map(LEM)*

Edge information is a useful object representation feature that is insensitive to illumination changes to certain extent. Though the edge map is widely used in various pattern recognition fields, it has been neglected in face recognition except in recent work. Edge images of objects could be used for object recognition and to achieve similar accuracy as gray-level pictures. A Line Edge Map approach, extracts lines from a face edge map as features. This approach can be considered as a combination of template matching and geometrical feature matching. The LEM approach not only possesses the advantages of feature-based approaches, such as invariance to illumination and low memory requirement, but also has the advantage of high recognition performance of template matching. Line Edge Map integrate the structural information with spatial information of a face image by grouping pixels of face edge map to line segments. After thinning the edge map, a polygonal line fitting process [62] is applied to generate the LEM of a face. An example of a human frontal face LEM is illustrated in Fig. 1. The LEM representation reduces the storage requirement since it records only the end points of line segments on curves. Also, LEM is expected to be less sensitive to illumination changes

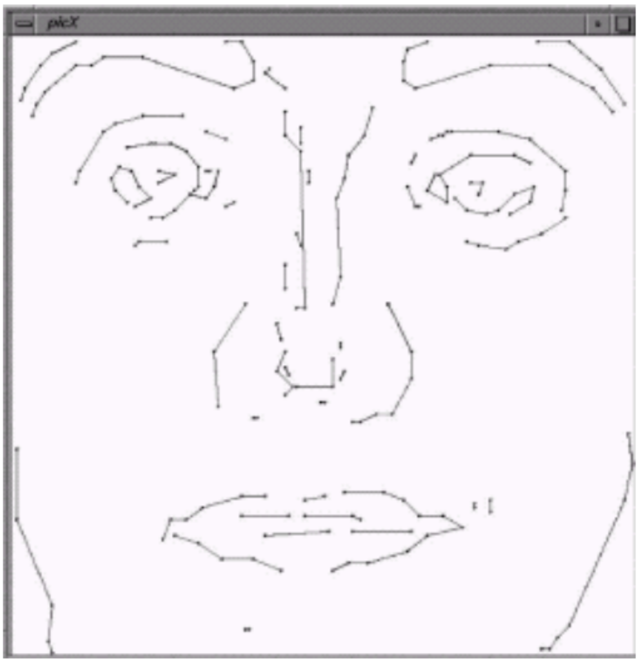

Fig. 1 An illustration of a face LEM

due to the fact that it is an intermediate-level image representation derived from low level edge map representation. The basic unit of LEM is the line segment grouped from pixels of edge map. A face prefilering algorithm is proposed that can be used as a preprocess of LEM matching in face identification application. The prefilering operation can speed up the search by reducing the number of candidates and the actual face (LEM) matching is only carried out on a subset of remaining models. Experiments on frontal faces under controlled /ideal conditions indicate that the proposed LEM is consistently superior to edge map. LEM correctly identify 100 percent and 96.43 percent of the input frontal faces on face databases [63,64], respectively. Compared with the eigenface method, LEM performed equally as the eigenface method for faces under ideal conditions and significantly superior to the eigenface method for faces with slight appearance variations (see Table I). Moreover, the LEM approach is much more robust to size variation than the eigenface method and edge map approach (see Table II) . In [61], the LEM approach is shown to be significantly superior to the eigenface approach for identifying faces under varying lighting condition. The LEM approach is also less sensitive to pose variations than the eigenface method but more sensitive to large facial expression changes.

3.2.3. Support Vector Machines(SVMs)

SVM is a learning technique that is considered an effective method for general purpose pattern recognition because of its high generalization performance without the need to add other knowledge . Intuitively, given a set of points belonging to two classes, a SVM finds the hyperplane that separates the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyperplane. This hyperplane is called Optimal Separating Hyperplane (OSH) which minimizes the risk of misclassifying not only the examples in the training set but also the unseen example of the test set. SVM can also be viewed as a way to train polynomial neural networks or Radial Basis function classifiers. The training techniques used here are based on the principle of Structure Risk Minimization (SRM), which states that better generalization capabilities are achieved through a minimization of the bound on the generalization error. Indeed, this learning technique is just equivalent to solving a linearly constrained Quadratic Programming (QP) problem. SVM is suitable for average size face recognition systems because normally those systems have only a small number of training samples. In summary, the main characteristics of SVMs are: (1) that they minimize a formally proven upper bound on the generalization error; (2) that they work on high-dimensional feature spaces by means of a dual formulation in terms of kernels; (3) that the prediction is based on hyperplanes in these feature spaces, which may correspond to quite involved classification criteria on the input data; and that outliers in the training data set can be handled by means of soft margins. The application of SVMs to computer vision problem have been proposed recently.

4. Experiments

Lots of technical jargon later, allow us to jump to our experiences and learnings through this project of ours.

The first step for us was to learn about convolutional neutral nets, a framework for making a face recognition model and some stuff about deep learning too. So , we learned this stuff from various articles and a bunch of You-Tube videos ,but our crewmate 'Parth' was a major help for us. He taught us 'Pytorch' from the very basic ,the code and its implementation by taking various examples . After getting the gist of 'Pytorch' , as a practice we tried the most basic MNIST dataset i.e. handwritten digits classification(0-9). Now, as the prerequisites were cleared, we decided to move on to our main problem statement i.e. face recognition. As face recognition consists of face detection and face identification , the next step was to detect the faces. For face detection, we used none other than "Haar-Cascade" which is maintained by the OpenCV library. So , now after detecting the face ,we wanted to just crop the face , for which we wrote a image pre-processed code. Now, we had finally obtained the image, whose array we will sending to the model. Initially when we started working on the project ,a huge hurdle for us was to select a proper dataset with a good set of images . We explored a lot on internet ,as the dataset we were looking for was neither meant to be too large(training would take quite an amount of time) nor meant to be too small(as it would overfit). Datasets like "CelebA", "VGG-Face2" , "Digi-Face" , etc, caught our eye ,as they had good amount of images per identity. As we had to start something at least to try, we selected the "Digi-Face" dataset. It consisted of 1.2 million images with around 1.1L identities and was recently released in 2022. We made a custom basic model and tried to train around 1 million images ,but as it was a bit too large ,the ram was getting filled and wasn't working properly , so we decided to take a quarter of the dataset which was trainable ,but would take some time. Actually, later we realised we did not have to import the complete dataset at once, instead we should have imported only the required number of images (i.e. images in one single batch) while training the model. There was another problem in the dataset which we realized a while later. The images in the dataset had some of the images in which the faces were looking sideways which would be a problem ahead. We then made a decision to create a custom dataset by taking images of 3 of us and 2 more hostel mates , each identity having around 20 images. We, pre-processed these images and the next thing was to create labels for them . So, we created the labels in the form of an array consisting of 0,1,2,3,4 as there were a total of 5 identities. Now, we had to make a model by ourselves, which was actually quite simple as we were made familiar with it in 'Pytorch' just a couple of days before. While making the model , we also took care that it should not have too many parameters. The next step was to create a training loop , which consisted of the loss and the optimizer steps. We used the Adam optimizer for the model. We trained the model with 5 epochs at start with a learning rate of 1e-3. For testing our model ,we just took a random photo of one of us and then passed the pre-processed one to the model, but we didn't get a correct prediction. We had some idea , why the prediction was coming out wrong. It was most probably due to overfitting, as the dataset was a little too small. So, to check it, we just sent an image which we used for training , to the model and the model predicted it perfectly. So ,now we came to know what exactly the problem was with the dataset.

5. Methodology

The solution we have arrived at is to use a pre-trained vgg16 model, trained on vggface dataset. We use this model to extract the features of a face, and then compare the features extracted with already extracted features of faces, using cosine similarity. Whatever face gives closest results, and if that result is satisfactory, it identiifies the face.

6. Summary and conclusions

We made a few models, and decided to take a pre-trained model vgg16 on vggface, converted the weights from tensorflow format to pytorch format, and implemented the model in our system.

Acknowledgements

Thanks to the article titled Face Recognition: A literature review by authors Ahmad Tolba, Ali El-Baz and Ahmad A El-Harby. Link: https://www.researchgate.net/publication/233864740

References