A COMPARISON OF THE DIFFERENT FACE - DETECTION ALGORITHMS AND THEIR IMPLEMENTATION

Priyanka Mohandas

ABSTRACT

This literature aims at reviewing the different face detection algorithms currently used in the field and compares their efficiency, execution time etc. An attempt has been made to choose the best algorithm based on the availability of the image processing libraries for use in ROS.

INTRODUCTION

Face detection is defined as the process of determining whether or not there are any faces in any given arbitrary image and, if present return the image location and extent of each face. The challenges associated with face detection are mainly due to the variability in scale, location, orientation (up-right, rotated), pose (frontal, profile), facial expression, occlusion and varying lighting conditions.

FACE DETECTION ALGORITHMS

The rapidly progressing research works on face processing methods to facilitate Human Computer Interaction (HCI) yielded several face detection methods during the past four decades. Among the face detection methods, the ones based on learning algorithms attract much attention today and they demonstrate excellent results. These data-driven methods rely heavily on the training sets. An important problem related to face detection is how to evaluate the performance of the proposed detection methods. For convenience, I am taking definitions from a survey paper by Ming-Hsuan Yang et.al [1]. Detection rate is defined as the ratio between the number of faces correctly detected and the number faces determined by a human. An image region identified as a face by a classifier is considered to be correctly detected if the image region covers more than a certain percentage of a face in the image. Detectors can make two types of errors: false negatives in which faces are missed resulting in low detection rates and false positives in which an image region is declared to be face, but it is not. A comprehensive evaluation should take these factors into consideration since one can adjust the parameters of one's method to increase the detection rates which can result in increasing the number of false detections which is undesirable.

Face detection can be viewed as a two-class recognition problem n which an image region is classified as being a face or non-face. Consequently, face detection is one of the few attempts to recognize from images (not abstract representations) a class of objects for which there is a great deal of within-class variability.

Face detection methods can be done classified as follows[1]:

1. Knowledge-based methods.
These rule-based methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face localization.

2. Feature invariant approaches.
These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization.

3. Template matching methods.
Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. These methods have been used for both face localization and detection.

4. Appearance-based methods.
In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

One problem with this approach is the difficulty in translating human knowledge into rules. If the rules are detailed (i.e., strict), they may fail to detect faces that do not pass all the rules. If the rules are too general, they may give many false positives. Moreover, it is difficult to extend this approach to detect faces in different poses since it is challenging to enumerate all the possible cases. On the other hand, heuristics about faces work well in detecting frontal faces in uncluttered scenes.

1. Knowledge Based Methods

In this approach, face detection methods are developed based on the rules derived from the researcher's knowledge of human faces. Simple rules are formulated to describe the features of a face and their relationships. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth. The relationships between features can be represented by their relative distances and positions.

2. Feature Based Model

In contrast to the knowledge-based approach, researchers have been trying to find invariant features of faces for detection. The hidden assumption is that humans can effortlessly detect faces and objects in different poses and lighting conditions and so, there must exist properties or features which are invariant over these variables. Many methods have been proposed to first detect facial features and then to find the presence of a face. Facial features such as eyebrows, eyes, nose, mouth, and hair-line are commonly extracted using edge detectors. Based on the extracted features, a statistical model is built to describe their relationships and to verify the existence of a face. One problem with the feature-based algorithms is that the image features can be severely corrupted due to illumination, noise, and occlusion. Features, skin colour and texture are the attributes of the images used for detection in this method.

3. Template Matching Methods

In template matching, a standard face pattern (usually frontal) is manually predefined or parameterized by a function. Given an input image, the correlation values with the standard patterns are computed for the face contour, eyes, nose, and mouth independently. The existence of a face is determined based on the correlation values. This approach has the advantage of being simple to implement. However, it has proven to be inadequate for face detection since it cannot effectively deal with variation in scale, pose, and shape .Multi-resolution, multi-scale, sub templates, and templates have subsequently been proposed to achieve scale and shape invariance.

4. Appearance Based Methods

In contrast to the template matching methods where the templates are predefined by experts, the templates in appearance-based methods are learned from the examples in the images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and nonface images. The learned characteristics are in the form of distribution models or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction is usually carried out for the sake of computation efficiency and detection efficacy.

Many appearance-based methods can be understood in a probabilistic framework. An image or feature vector derived from an image is viewed as a random variable x, and this random variable is characterized for faces and non-faces by the class-conditional density functions p(x| face) and p(x| non-face). Bayesian classification or maximum likelihood can be used to classify a candidate image location as face or non-face. Unfortunately, a straightforward implementation of Bayesian classification is infeasible because of the high dimensionality of x, because p(x| face) and p(x| non-face) are multimodal, and because it is not yet understood if there are natural parameterized forms for p(x| face) and p(x| non-face). Hence, much of the work in an appearance-based method concerns empirically validated parametric and nonparametric approximations to p(x| face) and p(x| non-face).

Another approach in appearance-based methods is to find a discriminant function (i.e., decision surface, separating hyper-plane, threshold function) between face and non-face classes .Conventionally, image patterns are projected to a lower dimensional space and then a discriminant function is formed (usually based on distance metrics) for classification, or a nonlinear decision surface can be formed using multilayer neural networks .Recently, support vector machines and other kernel methods have been proposed. These methods implicitly project patterns to a higher dimensional space and then form a decision surface between the projected face and non-face patterns.

Here, I discuss a few notable algorithms based on appearance based model.

4.1 Eigenfaces

The Eigenface approach began with a search for a low-dimensional representation of face images. Sirovich and Kirby [2] showed that Principal Component Analysis, also known as

Karhunen-Loeve method which is one of the popular methods for feature selection and dimension reduction, could be used on a collection of face images to form a set of basis features. These basis images, known as Eigenpictures, could be linearly combined to reconstruct images in the original training set. If the training set consists of *M* images, principal component analysis could form a basis set of *N* images, where $N < M$. The reconstruction error is reduced by increasing the number of eigenpictures, however the number needed is always chosen less than *M*. For example, if you need to generate a number of N eigenfaces for a training set of M face images, you can say that each face image can be made up of "proportions" of all this K "features" or eigenfaces:

Face image$_1$ = (23% of E_1) + (2% of E_2) + (51% of E_3) + ... + (1% E_n).

Recognition of human faces using PCA was first done by Turk and Pentland [3]. They expanded the results given by Sirovich and Kirby [2] and presented the Eigenface method of face recognition. In addition to designing a system for automated face recognition using eigenfaces, they showed a way of calculating the eigenvectors of a covariance matrix in such a way as to make it possible for computers at that time to perform eigen-decomposition on a large number of face images. Face images usually occupy a high-dimensional space and conventional principal component analysis was intractable on such data sets. Turk and Pentland's paper demonstrated ways to extract the eigenvectors based on matrices sized by the number of images rather than the number of pixels.

A set of eigenfaces is generated by performing PCA on a large set of images depicting different human faces. Informally, eigenfaces can be considered a set of "standardized face ingredients", derived from statistical analysis of many pictures of faces. Any human face can be considered to be a combination of these standard faces. For example, one's face might be composed of the average face plus 10% from eigenface 1, 55% from eigenface 2, and even -3% from eigenface 3. It does not take many eigenfaces combined together to achieve a fair approximation most faces.

Eigenface provides an easy and cheap way to realize face recognition in that:

- Its training process is completely automatic and easy to code.
- Eigenface adequately reduces statistical complexity in face image representation.
- Once eigenfaces of a database are calculated, face recognition can be achieved in real time.
- Eigenface can handle large databases.

However, the deficiencies of the eigenface method are also obvious:

- Very sensitive to lighting, scale and translation; requires a highly controlled environment.
- Eigenface has difficulty capturing expression changes.
- The most significant eigenfaces are mainly about illumination encoding and don't provide useful information regarding the actual face

4.2. Neural Networks

Neural networks have been applied successfully in many pattern recognition problems, such as optical character recognition, object recognition, and autonomous robot driving. Since face detection can be treated as a two class pattern recognition problem, various neural network architectures have been proposed. The advantage of using neural networks for face detection

is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one drawback is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance.

Among all the face detection methods that used neural networks, the most significant work is done by Rowley et al [5,6,7]. A multilayer neural network is used to learn the face and non-face patterns from face/non-face images (i.e., the intensities and spatial relationships of pixels). Sung [8] used a neural network to find a discriminant function to classify face and non-face patterns using distance measures .They also used multiple neural networks and several arbitration methods to improve performance. One limitation of the methods by Rowley et. al.[6] and by Sung [8] is that they can only detect upright, frontal faces.

4.3. Support Vector Machines

Support Vector Machines (SVMs) can be considered as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF) classifiers. While most methods for training a classifier (e.g., Bayesian, neural networks, and RBF) are based on of minimizing the training error, i.e., empirical risk, SVMs operates on another induction principle, called structural risk minimization, which aims to minimize an upper bound on the expected generalization error. An SVM classifier is a linear classifier where the separating hyperplane is chosen to minimize the expected classification error of the unseen test patterns. This optimal hyperplane isdefined by a weighted combination of a small subset of the training vectors, called support vectors. Estimating the optimal hyperplane is equivalent to solving a linearly constrained quadratic programming problem. However, the computation is both time and memory intensive.

4.4. Naive Bayes Classifier

Schneiderman and Kanade described a naive Bayes classifier to estimate the joint probability of local appearance and position of face patterns at multiple resolutions [9]. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes are much more distinctive than the pattern found around the cheeks .There are two reasons for using a naive Bayes classifier (i.e., no statistical dependency between the subregions) .First, it provides better estimation of the conditional density functions of these subregions .Second, a naive classifier provides a functional form of the posterior probability to capture the joint statistics of local appearance and position on the object .At each scale, a face image is decomposed into four rectangular subregions .These sub-regions are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns, and the statistics of each projected subregion are estimated from the projected samples to encode local appearance .Under this formulation, their method decides that a face is present when the likelihood ratio is larger than the ratio of prior probabilities. Schneiderman and Kanade later extend this method with wavelet representations to detect profile faces and cars [10].

4.5. Viola Jones Algorithm

In 2001 Paul Viola and Michael Jones proposed an object detection using Haar feature-based cascade classifiers in their paper [11]. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

It a visual object detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features and yields extremely efficient classifiers. The third contribution is a method for combining classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

The system yields face detection performance comparable to the best previous systems [5, 8, 10]. Implemented on a conventional desktop, face detection proceeds at 15 frames per second.

DISCUSSION

ALGORITHM SELECTION

According to a paper by Gregory Shakhnarovich et. al. [12], key competitors of the Viola Jones method include a neural network system, mentioned above, by Rowley et.al.[5,6,7] and a Bayesian system by Schneiderman and Kanade [9,10]. While the neural network system is broadly: considered to be the quickest previous system, the Viola-Jones system is slightly more accurate and ten times faster. Even though the Bayesian system has the highest reported detection rates, it is by far the slowest of the three. Hence, in terms of time taken for execution the above three algorithms can be represented as Bayesian model > Neural Network model > Viola-Jones model

METHOD	DETECTION RATE	COMPUTATIONAL AND STORAGE COSTS	SPEED	COMMENTS
Bayesian	HIGH	FIXED	LOW	probabilistic matching technique, better storage efficiency than Eigenfaces, uses PCA for reducing dimensions
Neural Network	HIGH	VARYING	MEDIUM	efficiency increases with the number of neuronal layers and nodes, adapts to the data
Viola-Jones	HIGH	FIXED	HIGH	uses Haar features, AdaBoost and casscade of classifiers

The characteristics of Viola–Jones algorithm which make it a good detection algorithm are:

- Robust – very high detection rate (true-positive rate) & very low false-positive rate always.
- Real time – For practical applications at least 2 frames per second must be processed.
- Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process).

The advantages of using Viola-Jones method are as listed below

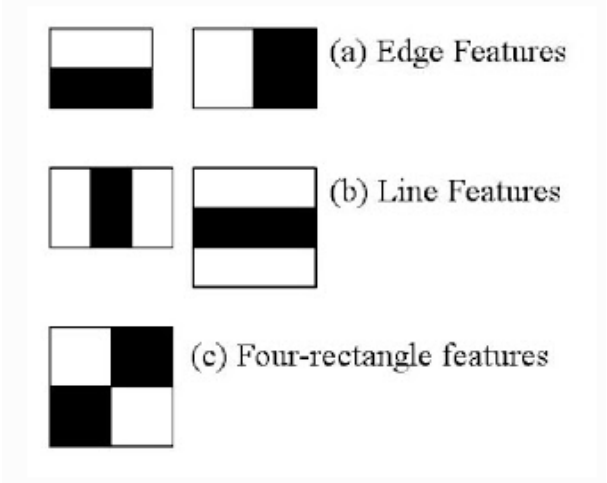
- Extremely fast feature computation
- Efficient feature selection
- Scale and location invariant detector
- Instead of scaling the image itself (e.g. pyramid-filters), we scale the features.
- Such a generic detection scheme can be trained for detection of other types of objects (e.g. cars, hands)

The Viola –Jones model, which created a breakthrough in face detection, has been observed to yield impressive results. This approach is now the most commonly used algorithm for face detection

Hence, I choose the Jones-Viola method for face detection in this exercise.

Now that we have chosen the algorithm, let us how this method works A simple intuitive explanation of how Viola-Jones face detection works is explained below.

1. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then features are to be extracted from it. For this, Haar features shown in the figure below are used. They are just like the convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.



2. All possible sizes and locations of each kernel are used to calculate plenty of features. In a standard 24x24 pixel sub-window, there are a total of 162336 possible features, and for each feature calculation, we need to find sum of pixels under white and black rectangles. It would be prohibitively expensive to evaluate them all when testing an image. To solve this, the concept of integral images is introduced. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. The integral image representation evaluates rectangular features in constant time, which gives them a considerable speed advantage over more sophisticated alternative features. Because each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in nine. The integral image at location (x,y), is the sum of the pixels above and to the left of (x,y), inclusive.

3. But among all these features, most of them are irrelevant. The best features out of 160000+ features is achieved by a machine learning algorithm called AdaBoost. The object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

For this, each and every feature is applied on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. The features with minimum error rate are selected, which means they are the features that best classifies the face and non-face images. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates and weights are calculated. The process is continued until required accuracy or error rate is achieved or till the required number of features are found.

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can’t classify the image, but together with others forms a strong classifier. The paper[11] says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features.

4. So, now if an image is taken, and each 24x24 window is considered, applying 6000 features to it to check if it is face or not, is a little inefficient and time consuming. Viola and Jones solved this problem by introducing the concept of Cascade Of Classifiers. Instead of applying all the 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, it is discarded. The remaining features are not considered, on it. If it passes, the second stage of features are applied and the process is continued. The window which passes all stages is a face region.

The original detector, as in the paper[12], had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. According to its authors, on an average, 10 features out of 6000+ are evaluated per sub-window.

IMPLEMENTATION SELECTION

Various improvements on the Viola-Jones method have been made since its inception. Its implementation is mainly done using OpenCV and MATLAB.

1. OpenCV Implementation

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python.

It comes with a trainer as well as detector. OpenCV can be used to create your own trainer for your own classifier for any object like car, planes etc.

Though the theory may sound complicated, in practice it is quite easy. The cascades themselves are just a bunch of XML files that contain OpenCV data used to detect objects. You initialize your code with the cascade you want, and then it does the work for you. Since face detection is such a common case, OpenCV comes with a number of built-in cascades for detecting everything from faces to eyes to hands and legs.

2. MATLAB Implementation

In MATLAB, the cascade object detector uses the Viola-Jones algorithm to detect people's faces, noses, eyes, mouth, or upper body. The Training Image Labeler can be used to train a custom classifier to use with this System object.

However, due to the availability of OpenCV libraries as open source entities, they are preferred over MATLAB.

There are a few open source libraries such as fdlibs which provide face detection support in C/C++. Python is easy to learn, easy and fast to code, and more readable than C. Hence, due to the ease of programming and fewer lines of code needed to implement face detection using Viola – Jones method using Python in OpenCV, I choose OpenCV libraries to implement face detection.

REFERENCES

[1]. Yang, Ming-Hsuan; J Kreigman, David; Ahuja, Narendra " Detecting Faces In Images: A Survey". IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 24, no.1, January 2002

[2].M. Kirby; L. Sirovich "Application of the Karhunen-Loeve procedure for the characterization of human faces". IEEE Transactions On Pattern Analysis and Machine Intelligence vol .12, no .1, pp .103-108, January1990.

[3] M .Turk and A .Pentland, "Eigenfaces for Recognition". Journal Of Cognitive Neuroscience, vol .3, no .1, pp .71-86, 1991.

[4]. Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems". Annals of Eugenics 7 (2): 179–188, 1936.

[5]. H .Rowley, S .Baluja, and T .Kanade. "Human Face Detection inVisual Scenes".Advances in Neural Information Processing Systems 8,D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds., pp. 875-881, 1996.

[6]. H .Rowley, S .Baluja, and T .Kanade. "Neural Network-Based Face Detection". Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp .203-208, 1996 .

[7]. H .Rowley, S .Baluja, and T .Kanade." Neural Network-Based Face Detection". IEEE Trans. Pattern Analysis and Machine Intelligence, vol .20, no .1, pp .23-38, Jan .1998

[8]. K.-K. Sung. "Learning and Example Selection for Object and Pattern Detection". PhD thesis, Massachusetts Inst .of Technology, 1996

[9]. H .Schneiderman and T .Kanade. "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition". Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp .45-51, 1998.

[10] H .Schneiderman and T .Kanade. "A Statistical Method for 3D Object Detection Applied to Faces and Cars". Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol .1, pp .746-751, 2000 .

[11]. Paul Viola and Michael J. Jones. "Robust real-time object detection". Proc. of IEEE Workshop on Statistical and Computational Theories of Vision, 2001.

[12].G. Shakhnarovich, P.A. Viola, B. Moghaddam. "A Unified Learning Framework for Real Time Face Detection and Classification". Proceedings. Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002.