A Neural Network Based Face Detection Approach

Anjana Mall #1, Mrs. Shusmita Ghosh*2

#Deptt. Of Electronics & Telecommunication, CSVTU, Bhilai
Chhattisgarh, Bhilai - India
1anjana.mll@gmail.com
2rinkgh@gmail.com

Abstract—Object detection is a fundamental problem in computer vision. For such applications as image indexing, simply knowing the presence or absence of an object is useful. Detection of faces, in particular, is a critical part of face recognition and, and critical for systems which interact with users visually. In this paper we are giving neural network based approach for face detection.

Key Word: Feature Based, YChCr, Backpropogation.

I. INTRODUCTION

Pattern recognition is a modern day machine intelligence problem with numerous applications in a wide field, including Face recognition, Character recognition, Speech recognition. The field of pattern recognition is still very much in it's infancy, although in recent years some of the barriers that hampered such automated pattern recognition systems have been lifted due to advances in computer hardware providing machines capable of faster and more complex computation. Face recognition, although a trivial task for the human brain has proved to be extremely difficult to imitate artificially. It is commonly used in applications such as human-machine interfaces and automatic access control systems. Face recognition involves comparing an image with a database of stored faces in order to identify the individual in that input image. The related task of face detection has direct relevance to face recognition because images must be analyzed and faces identified, before they can be recognized. Detecting faces in an image can also help to focus the computational resources of the face recognition system, optimizing the systems speed and performance. Face detection involves separating image windows into two classes; one containing faces (targets), and one containing the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. For basic pattern recognition systems, some of these effects can be avoided by assuming and ensuring a uniform background and fixed uniform lighting conditions. This assumption is acceptable for some applications such as the automated separation of nuts from screws on a production line, where lighting conditions can be controlled, and the image background will be uniform. For many applications however, this is unsuitable, and systems must be designed to accurately classify images subject to a variety unpredictable conditions.

Fig. 1 Face Detection System

II. OVERVIEW OF FACE DETECTION TECHNIQUES

Low [1] conducted a survey on face detection techniques, and identified two broad categories that separate the various approaches, appropriately named feature-based and image-based approaches. Each category will be explained, and the work completed will be presented, providing a brief yet thorough overview of the various face detection techniques.

A. Feature-Based Approach:

Low [1] divides the group of feature-based systems into three subcategories: low-level analysis, feature analysis, and active shape models.

A1. Low-level Analysis

Low-level Analysis deals with the segmentation of visual features using the various properties of the pixels, predominantly gray-scale or color. Edge representation (detecting changes in pixel properties) was first implemented by Sakai et al [2] for detecting facial features in line drawings. Craw et al [3] developed this further to trace a human head outline, allowing feature analysis to be constrained to within the head outline. Various operators are used to detect the presence of an edge, including the Sobel operator, the Marr-Hildreth operator, and a variety of first and second derivatives of Gaussians. All edge-based techniques rely on labeled edges which are matched to a face model for verification. Govindaraju [4] labeled edges as left side, hairline, and right side, developing a system capable of detecting 76% of faces in a set of 60 images with complex backgrounds, with an average of two false alarms per image. Gray Information can be used to identify various facial features. Generally Eyebrows, pupils and lips appear darker than surrounding regions, and thus extraction algorithms can search for local minima. In contrast, local maxima can be used to indicate the bright facial spots such as nose tips [5]. Detection is then performed using low-level GRAY-scale thresholding. Yang and Huang [6] explore grayscale behavior
using an image pyramid (similar to that used later in the project), where an image is resized. Using the hypothesis that at low resolutions face regions will become uniform, face candidates are established which are verified by the existence of local minima at high resolution, representing facial features.

Color contains extra dimensions which can help differentiate two regions which may contain similar gray information but appear very different in color space. It was found that different skin color gives rise to a tight cluster in color space, thus color composition of human skin differs little across individuals, regardless of race. The most widely used color model is RGB, although there are many others that exist and have been used (a comparison can be found in Terrillon et al [7]). Motion Information (where available) can be used to assist in the detection of human faces, using the principle that, if using a fixed camera, the “background clutter” will remain somewhat static, relative any “moving object”. A straightforward way to achieve motion segmentation is by frame difference analysis. Thresholding accumulated frame differences is used to detect faces or facial features. Another way to measure motion is through the estimation of moving image contours, a technique that has proven to be more reliable, particularly when the motion is insignificant.

A2. Feature Analysis

Low-level analysis introduces ambiguity which can be solved by high level feature analysis, often through the use of some additional knowledge about the face. There are two approaches for the application of this additional knowledge (commonly face geometry). The first involves sequential feature searching strategies based on the relative positioning of individual facial features. Initially prominent facial features are determined which allow less prominent features to be hypothesized (for example a pair of dark regions found in the face area increases the confidence of facial existence). The facial feature extraction algorithm [8], is a good example of feature searching, achieving 82% accuracy with invariance to gray and color information, failing to detect faces with glasses and hair covering the forehead. A similar system proposed by Jeng et al [9] reported an 86% detection rate.

The second technique, constellation analysis, is less rigid and is more capable of locating faces of various poses in complex backgrounds. It groups facial features in face-like constellations, using robust modeling methods such as statistical analysis. Burl et al [10] use statistical shape theory on features detected from a multi-scale Gaussian derivative filter, capable of detecting 84% of faces, with some invariance to missing features, translation, rotation and scale. Probabilistic face models based on multiple face appearance have also been used in many systems including Yow and Cipolla’s model [11] reporting a 92% detection rate.

A3. Active Shape Model

Active shape models represent the actual physical and hence higher-level appearance of features. These models are released near to a feature, such that they interact with the local image, deforming to take the shape of the feature. There are three types of active shape models that have been used throughout the literature: snakes, deformable templates and smart snakes. Snakes, or Active Contours, are commonly used to create a head boundary. Created nearby, they lock on to nearby edges, eventually assuming the shape of the head. The evolution of a snake is achieved by minimizing an energy function, which consists of the sum of an internal energy function, defining its natural evolution (typically shrinking or expanding), and an external energy function, which counteracts the internal energy enabling the contours to deviate from the natural evolution. Energy minimization can be obtained by optimization techniques such as the steepest gradient descent although the additional computational demands have encouraged others to use faster iteration methods [12], and Lam and Yan [13]. Snakes are prone to becoming trapped on false image features and are not efficient in extracting non-convex features due to their tendency to attain minimum curvature. Gunn and Nixon [14], introduced a parameterized snake model to overcome these limitations by using dual integrated snakes. Deformable Templates can be used as an extension to the snake models. Yuille et al [15] incorporated global information of the eye to improve the extraction process, using a deformable eye template. Once established near an eye feature, would deform itself, toward optimal feature boundaries using steepest gradient descent minimization as the deformation mechanism. One limitation of such techniques is that they are sensitive to initial placement. Yuille et al [15] showed promising results when the template was placed below the eye, but noted that if place above the eye, the template may be attracted towards the eyebrow instead. A further limitation is the processing time attempts have been made to improve this. Besides eye templates, the use of mouth templates has also been introduced. Smart Snakes, or Point Distributed Models (PDMs) are compact parameterized descriptions of a shape based upon statistics. They use Principle Components Analysis (PCA) to construct a linear flexible model from variations of the features in a training set. Face PDM was first developed by Lantis et al[16] as a flexible model with promising results (95% detection rate). My system is based on skin color Shuangbao Sho et.al[31] based which would first segment the skin based on YCbCr model and then many face area are trained using neural network backpropagation algorithm.

Image-Based Approach:

Face detection by explicit modeling of facial features is a very rigid approach which has been shown to be troubled by the unpredictability of faces and environmental conditions. There is a need for more robust techniques, capable of performing in hostile environments, such as detecting multiple faces with clutter-intensive backgrounds. This has inspired a new research area in which face detection is treated as a general pattern recognition problem. Whereas face recognition deals with recognizing the face, face detectors must recognize an object as
a face, from examples. This eliminates the problem of potentially inaccurate models based on erroneous or incomplete face knowledge and instead places the emphasis on the training examples from which the system which learn to distinguish a face. Most image-based approaches apply a window scanning technique for detecting faces, which due to its exhaustive nature, increases computational demand.

B1 Linear Subspace methods

Images of human faces lie in a subspace of overall image space which can be represented by methods closely related to standard multivariate statistical analysis, including Principal Component Analysis (PCA), Linear Discriminate Analysis (LDA), and Factor Analysis (FA). In the late 1980s, Sirovich and Kirby [17] developed a technique using PCA to represent human faces. The technique first finds the principal components of the distribution of faces (expressed in terms of eigenvectors). Each face in the set can then be approximated by a linear combination of the largest eigenvectors, more commonly referred to as eigenfaces. Moghaddam and Pentland [18] later proposed a facial feature detector using ‘Distance From Space Face’ (DFFS – a measure of faceness), generated from eigenfeatures (eigeneyes, eigennose, eigenmouth), obtained from various facial feature templates in a training set. The performance of the eye locations was reported to be 94% with only 6% false positive (uniform background).

Various improvements to this simple system were made, notably that of Sung and Poggio [19] which represented the face space by dividing it into subclasses. Their detector (detailed in section 2.2.2) involves four steps: image pre-processing, distribution-based model construction, image to model measurements taken, Classification. One issue arising from the application of such systems is the problems associated with the collection of a representable set of training samples, particularly for non-face images. Sung and Poggio [19] suggest the use of a bootstrap algorithm, now widely implemented, which involves adding false detections to the training set as non-face examples. Yang et al [20] proposed a method for face detection based on Factor Analysis (FA), which is similar to PCA but assumes the observed data samples come from a well defined model. Using a mixture of factor analyzers, a set of training images is used to estimate the parameters in the mixture model. This model is then applied to sub windows in the input image, and the probability of a face being present is returned. Yang et al also proposed a system using Kohonen’s Self Organising Maps [21], and an LDA. Whereas PCA is aimed at representation, LDA aims for discrimination and is therefore well suited to face detection when the class of faces and non-faces is divided into subclasses.

B2. Neural Networks

Early approaches based on the simple Multiple Layer Perceptrons (MLP) gave encouraging results on fairly simple datasets. The first advanced neural approach which reported performance statistics on a large, visually complex dataset, was by Rowley et al [22]. Their system incorporates face knowledge in the rationally connected neural network architecture, with specialized window sizes designed to best capture facial information (e.g. horizontal strips to identify the mouth). Images are pre-processed before being classified by the network, the output from which is post-processed to remove overlapping detections, resulting in one detection per face, and a reduction in false positives. Multiple networks were trained independently and their outputs combined using various arbitration methods to further improve performance. Their detector produced very impressive results, and has since been incorporated in many other systems including several commercial applications. A different neural approach was suggested by Feraud et al [23] based on a Constrained Generative Model (CGM), an auto associative fully connected MLP with three layers. Lin et al [24], propose a fully automatic face recognition system based on probabilistic decision-based neural networks (PDBNN). For face detection the network has only one subnet representing the face class, and training is unsupervised.

B3. Statistical Approaches

Systems based on information theory, support vector machines and Bayes’ Decision Rule are examples of Image-Based approaches that do not fit into either of the other categories. Colmenarez and Huang [25] developed a system based on Kullback relative information. This divergence is a non-negative measure of the difference between two probability density functions. During training, for each pair of pixels in the training set, a joint-histogram is used to create probability functions for the classes of faces and non-faces. The training procedure uses a large quantity of 11x11 pixel images and results in a set of look-up tables of likelihood ratios. Pairs of pixels which contribute poorly to detection are completely removed from the look-up tables to reduce computational requirements. The system was further improved by incorporating a bootstrap training algorithm. In Osuna et al [26] a support vector machine (SVM) is applied to face detection. A SVM with a 2nd degree polynomial as a kernel function is trained with a decomposition algorithm which guarantees global optimality. Images are pre-processed and trained with a bootstrap learning algorithm. Other research into SVMs has attempted to improve the Osuna detector [26] with good results [27]. Schneider man and Kanade [28, 29] propose two face detectors based on Bayes’ decision rule.

\[ P(\text{image} | \text{object}) > P(\text{non-object}) \]
\[ P(\text{image} | \text{non-object}) > P(\text{object}) \]

A face exists at the current location if the above condition is true. The Bayes Decision rule is proven to give the optimal solution providing \( P(\text{image} | \text{object}) \) and \( P(\text{image} | \text{non-object}) \) are accurate. Schneider man and Kanade [29] implement a wavelet transform in their second system which decomposes the image into 10 sub bands, from which 17 visual attributes are extracted, and treated as statistically independent random variables.
III. PROPOSED FACE DETECTION APPROACH

The system operates in two stages: it first applies a set of neural network-based detectors to an image, and then uses an arbitrator to combine the outputs. The individual detectors examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual networks and eliminates overlapping detections. The first component of our system is a neural network that receives as input a $20 \times 20$ pixel region of the image, and generates an output ranging from 1 to -1, signifying the presence or absence of a face, respectively. To detect faces anywhere in the input, the network is applied at every location in the image. To detect faces larger than the window size, the input image is repeatedly reduced in size (by sub sampling), and the detector is applied at each size. This network must have some invariance to position and scale. The amount of invariance determines the number of scales and positions at which it must be applied. For the work presented here, we apply the network at every pixel position in the image, and scale the image down by a factor of 1.2 for each step in the pyramid.

![Image of network structure](image)

**Fig. 2** The basic algorithm used for face detection.

After a $20 \times 20$ pixel window is extracted from a particular location and scale of the input image pyramid, it is preprocessed using the affine lighting correction. The preprocessed window is then passed to a neural network. The network has retinal connections to its input layer; the receptive fields of hidden units are shown in Figure 2. The input window is broken down into smaller pieces, of four $10 \times 10$ pixel regions, sixteen $5 \times 5$ pixel regions, and six overlapping $20 \times 5$ pixel regions. Each of these regions will have complete connections to a hidden unit, as shown in the figure. Although the figure shows a single hidden unit for each sub region of the input, these units can be replicated. For the experiments which are described later, we use networks with two and three sets of these hidden units. The shapes of these sub regions were chosen to allow the hidden units to detect local features that might be important for face detection. In particular, the horizontal stripes allow the hidden units to detect features such as mouths or pairs of eyes, while the hidden units with square receptive fields might detect features such as individual eyes, the nose, or corners of the mouth. Other experiments have shown that the exact shapes of these regions do not matter; however it is important that the input is broken into smaller pieces instead of using complete connections to the entire input. Similar input connection patterns are commonly used in speech and character recognition tasks [Waibel et al., 1989, Le Cun et al., 1989]. The network has a single, real-valued output, which indicates whether or not the window contains a face.

A. Face Training Images

In order to use a neural network to classify windows as faces or no faces, we need training examples for each set. For positive examples, the faces are scaled to a uniform size, position, and orientation within a $20 \times 20$ pixel window. The images are scaled by a random factor between $1/\sqrt{1.2}$ to $\sqrt{1.2}$, and translated by a random amount up to 0.5 pixels. This allows the detector to be applied at each pixel location and at each scale in the image pyramid, and still detect faces at intermediate locations or scales. In addition, to give the detector some robustness to slight variations in the faces, they are rotated by a random amount (up to $10^\circ$ degrees). In our experiments, using larger amounts of rotation to train the detector network yielded too many false positives to be usable. There are a total of 1046 training examples in our training set, and 15 of these randomized training examples are generated for each original face. The next sections describe methods for collecting negative examples and training.

B. Non-Face Training Images

We needed a large number of no face images to train the face detector, because the variety of no face images is much greater than the variety of face images. One large class of images which do not contain any faces are pictures of scenery, such as trees, mountains, and buildings. There is a large collection of images located at [http://wuarchive.wustl.edu/multimedia/images/gif/](http://wuarchive.wustl.edu/multimedia/images/gif/). We selected the images with the keyword “Scenery” in their descriptions from the index, and downloaded those images. This, along with a couple other images from other sources, formed our collection of 120 non face “scenery” images. Collecting a “representative” set of no faces is difficult. Practically any image can serve as a no face example; the space of non face images is much larger than the space of face images. The statistical approach to machine learning suggests that we should train the neural networks on precisely the same distribution of images which it will see at runtime. For our face detector, the number of face examples is 15,000, which is a practical number. However, our representative set of scenery images contains approximately 150,000,000 windows, and training on a database of this size is very difficult. The next two
sections describe two approaches to training with this amount of data.

C. Active Learning

Because of the difficult of training with every possible negative example, we utilized an algorithm described in [Sung, 1996]. Instead of collecting the images before training is started, the images are collected during training, in the following manner:

1. Create an initial set of non face images by generating 1000 random images. Apply the preprocessing steps to each of these images.

2. Train a neural network to produce an output of 1 for the face examples, and -1 for the non face examples. On the first iteration of this loop, the network’s weights are initialized randomly. After the first iteration, we use the weights computed by training in the previous iteration as the starting point.

3. Run the system on an image of scenery which contains no faces. Collect sub images in which the network incorrectly identifies a face (an output activation > 0).

4. Select up to 250 of these sub images at random, apply the preprocessing steps, and add them into the training set as negative examples. Go to Step 2.

The training algorithm used in Step 2 is the standard error back propagation algorithm with a momentum term [Hertz et al., 1991]. The neurons use the than activation function, which gives an output ranging from $-1$ to $1$, hence the threshold of 0 for the detection of a face. Since we are not training with all the negative examples, the probabilistic arguments of the previous section do not apply for setting the detection threshold.

Since the number of negative examples is much larger than the number of positive examples, uniformly sampled batches of training examples would often contain only negative examples, which would be inappropriate for neural network training. Instead, each batch of 100 positive and negative examples is drawn randomly from the entire training sets, and passed to the back propagation algorithm as a batch. We choose the training batches such that they have 50% positive examples and 50% negative examples. This ensures that initially, when we have a much larger set of positive examples than negative examples, the network will actually learn something from both sets. Note that this changes the distribution of faces and non faces in the training sets compared with what the network will see at run time. Although theoretically the wrong thing to do, [Lawrence et al., 1998] observes such techniques often work well in practice.
V. CONCLUSION

Face detection has many applications including Security applications. Until recently, much of the work in the field of computer vision has focused on face recognition, with very little research into face detection. Human face detection is often the first step in the recognition process as detecting the location of a face in an image, prior to attempting recognition can focus computational resources on the face area of the image. Although a trivial task that human performs effortlessly, the task of face detection is a complex problem in computer vision, due to the great multitude of variation present across faces. Several techniques have been proposed to solve this problem (as discussed in the Literature Survey, section 2), including what. Low [1] term Feature-Based approaches, and the more recent Image-Based approaches. Both categories of approaches offer systems worthy of recognition with promising results. Feature-Based Approaches, often applied to real-time systems, are often reliant on a priori knowledge of the face, which is used explicitly to detect features. The more robust Image-Based approaches are too computationally expensive for real-time systems, although systems using a hybrid of both approaches are being developed with encouraging results. Arguably the stronger of the two approaches, image-based systems are being increasingly investigated, with much of the work involved in overcoming the limitations of computation time. Sanner’s detector [21], evaluated throughout this project, is an example of an image-based approach, based on the Rowley et al [22] detector, which has helped set the standard to which others must comply. Sanner’s detector implements some other key features of the Rowley et at detector [22], draws from the work of Sung and Poggio [19], and has been shown to produce encouraging results, despite the limitations stated. The changes made to the detector improved the detection rate, at the expense of a greater number of false positives. It would depend on the application, as to whether this was considered an actual improvement in performance or not, as some applications would sacrifice the detection of a few faces to keep the number of false positives as low as possible, whereas others would prefer to maximize the number of face detection regardless (e.g. a security system that identifies possible faces to be cross-checked with a database of existing offenders would rather ‘waste time’ processing false positives, than miss checking a true detection). For many systems though this somewhat primitive implementation would be unsuitable and would require extensive improvement.

REFERENCES