Neural Network Based Face Detection

Mamta Pradhan
Department of Biomedical Engineering
National Institute of Technology,Raipur

November 2021

1 Abstract

In this term paper discuss about the neural network-based face detection system means how to detect upright, frontal views of faces in gray scale pictures using a neural network based method. The method works by resolving the outputs of one or more neural networks applied directly to parts of the input picture. Each network is programmed to output whether a face is there or not. The algorithms and training techniques are intended to be universal. Face modification is possible. Firstly brief introduction about the neural network and conventional method of face detection, after that brief description about the systems and neural network based filter and merging overlapping detections and arbitration sensitivity analysis third I discuss about the improving the result and after that conclusion and future scope of this neural network based face detection.

2 Introduction

An artificial neural network is an attempt to simulate the network of neurons that make up a human brain so that the computer will be able to learn things and make decisions in a human-like manner. Many face detection researchers have used the idea that facial images can be characterized directly in terms of pixel intensities. These images can be characterized by probabilistic models of the set of face images or implicitly by neural networks. The parameters for these models are adjusted either automatically from example images or by hand. A few researchers have taken the approach of extracting features and applying either manually or automatically generated rules for evaluating these features.

Conventional Method of Face Detection

In past different type of machine learning based method is used to detect the face. There are three type of conventional (Based on machine learning) method is used in face detection.

(A) The knowledge-based (Rule Based/Top Down) - This method is depends on the set of rules, and it is based on human knowledge to detect the faces. For example a face must have a nose, eyes, and mouth within certain distances and positions with each other.

Advantages

1) Reduced computational complexity.

Disadvantages

1)The big problem with these methods is the difficulty in building an appropriate set of rules. There could be many false positive if the rules were too general or too detailed. This approach alone is insufficient and unable to find many faces in multiple images.

2)Low detection rates

3)Difficulty to detect face in complex back grounds or in presence of multiple face.

4)Application only for frontal faces. May not detect faces with beards.

(B) Feature-Based:- The feature-based method is to locate faces by extracting structural features of the face. It is first trained as a classifier and then used to differentiate between facial and non-facial regions. The idea is to overcome the limits of our instinctive knowledge of faces. This approach divided into several steps and even photos with many faces they report a success rate of 94

The feature based detection is divided into three part that is facial feature, skin feature and texture based.

Advantages of facial features based

Reduces computational costs and computational complexity for feature representation.

Disadvantages of facial features based

1) Convolution cost is very high.

2) It has a high dimensional feature vector Advantages of skin colour features based:- Fast, robust and also have a high detection rate.

Disadvantages of skin color features based

Sensitive to illumination, face sizes , poses and expressions.

Advantages of texture

Tolerance to monotonic illumination changes and computational simplicity.

Disadvantages of texture based

unable to detect the faces in occlusions and different poses.

(C) Template Matching

Template Matching method uses pre-defined face templates to locate or detect the faces by the correlation between the templates and input images. Ex- a human face can be divided into eyes, face contour, nose, and mouth. Also, a face model can be built by edges just by using edge detection method.

Advatnges of template based method

Demonstrates goods performance in tracking non rigid features and this approach is simple to implement.

Disadvatnges of template based method

but it is inadequate for face detection. However, deformable templates have been proposed to deal with these problems like scale pose and shape.

Disadvantages of this overall this method is

1) Massive data storage burden. The ML technology used in face detection requires powerful data storage that may not be available to all users.

2) These algorithms were heavily affected by factors such as extreme head poses (where the head is rotated far to one side or tilted far up or far down, for example) and varying lighting conditions.

In Appearance Based

The appearance-based method includes neural network method and it depends on a set of delegate training face images to find out face models. The appearance-based approach is better than other ways of performance. In general appearance-based method rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. we can use neural network methods to carry out accurate face detection in a wide range of scenarios like extreme head poses (where the head is rotated far to one side or tilted far up or far down, for example) and varying lighting conditions.

The two classes to be discriminated in face detection are “images containing faces” and “images not containing faces.” It is easy to get a representative sample of images which contain faces, but much harder to get a representative sample of those which do not. We avoid the problem of using a huge training set for non-faces by selectively adding images to the training set as training progresses. This “bootstrap” method reduces the size of the training set needed. The use of arbitration between multiple networks and heuristics to clean up the results significantly improves the accuracy of the detector.

3 Description about the sytem

Our system operates in two stages: It first applies a set of neural network-based filters to an image and then uses an arbitrator to combine the outputs. The filters examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual filters and eliminates overlapping detections.

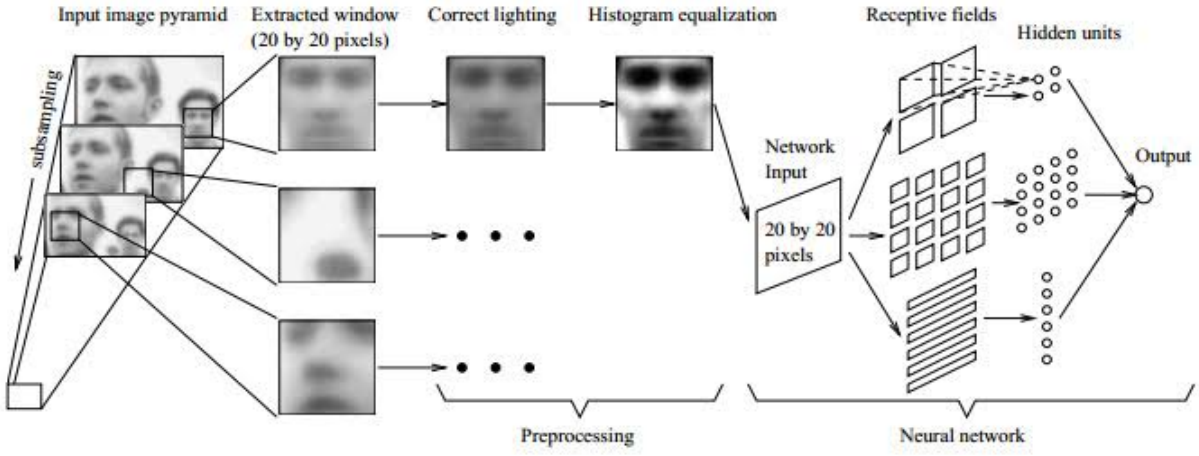

Figure 1: The basic algorithm used for face detection

4 Stage One: A Neural Network-Based Filter

The first component of our system is a filter that receives as input a 20 * 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 filter 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 subsampling), and the filter is applied at each size. This filter 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 filter at every pixel position in the image and scale the image down by a factor of 1.2 for each step in the pyramid.

The filtering algorithm reveals information. A preprocessing step technique is first applied to a portion of the image window. After then, the window is processed through a neural network, which determines if it includes a face. The preprocessing tries to balance the intensity values throughout the window at first. The intensity values in an oval region inside the window were fitted using a function that changes linearly throughout the window. Because pixels outside the oval may represent the backdrop, the illumination variance across the face is computed without them. The linear function can be subtracted from the window to accommodate for a range of lighting situations by approximating the total brightness of each section of the window. After that, histogram equalization is used, which nonlinearly maps the intensity data to broaden the window's range of intensities. For pixels within an oval section in the window, the histogram is computed. This adjusts for variations in camera input gains and, in some circumstances, improves contrast. After that, the preprocessed window is sent into a neural network. The hidden units' receptive fields have retinal connections to the network's input layer. There are three sorts of hidden units: four that look at 10 * 10 pixel subregions, sixteen that look at 5 * 5 pixel subregions, and six that look at overlapping horizontal stripes of pixels of 20 * 5 pixels. Each of these kinds was chosen to enable the concealed units to detect local characteristics that may be useful for face detection.

Hidden units with horizontal stripes may identify characteristics like lips or pairs of eyes, whereas hidden units with square receptive fields can detect features like individual eyes, the nose, or the corners of the mouth. Although each subregion of the input is represented by a single hidden unit in the diagram, these units can be duplicated. We employ networks with two and three sets of these hidden units in the studies that follow. In speech and character recognition tasks, similar input connection patterns are frequently employed. The network has a single real-valued output that specifies whether the window has a face or not. And scale, which results in multiple boxes around some faces. Note also that there are some false detections; they will be eliminated by methods presented in Section 5.

A significant number of face and nonface pictures are required to train the neural network utilised in stage one to act as an accurate filter. Nearly 1,050 face samples were culled from face databases at CMU and Harvard2 as well as the internet. Faces of various sizes, orientations,

positions, and intensities were included in the images. Each face's eyes, tip of nose, corners, and centre of mouth were manually labelled. These points were utilised to scale, orient, and position each face to the same scale, as seen below.

1) Initialize F , a vector which will be the average positions of each labeled feature over all the faces, with the feature locations in the first face F1.

2) The feature coordinates in F are rotated, translated, and scaled, so that the average locations of the eyes will appear at predetermined locations in a 20 ¥ 20 pixel window.

3) For each face i, compute the best rotation, translation, and scaling to align the face's features Fi with the average feature locations F . Such transformations can be written as a linear function of their parameters. Thus, we can write a system of linear equations map ping the features from Fi to F . The least squares solution to this over constrained system yields the parameters for the best alignment transformation. Call the aligned feature locations F¢i.

4) Update F by averaging the aligned feature locations Fi ¢ for each face i.

Within five rounds, the alignment algorithm converges, giving a function for each face that maps that face to a 20 × 20 pixel window. Each original picture is randomly rotated (around its centre points) up to 10 degree scaled between 90 and 110 percent, translated up to half a pixel, and mirrored to produce fifteen face samples for the training set. After that, each of the 20 20 windows in the set is preprocessed (by applying lighting correction and histogram equalization). The filter is invariant to translations of less than a pixel and scalings of 20

Because the universe of nonface pictures is significantly bigger than the space of face images, practically any image may be used as a nonface example. Collecting a "representative" group of nonfaces, on the other hand, is challenging.Rather than gathering photos before beginning training, images are taken throughout training.

1) Create an initial set of no face images by generating 1,000 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. The training algorithm is standard error back propagation with momentum. 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 in correctly 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.

It's worth noting that several of the instances resemble faces, albeit they're not as similar as the positive ones. Because of these examples, the neural network is forced to learn the exact difference between face and non face images. In the bootstrap method mentioned above, we utilised 120 photos of landscapes to collect negative instances. A typical training run picks around 8,000 nonface pictures from the 146,212,178 sub images accessible in the training scenery photographs at all scales and locations. [5] proposed a similar training technique in which a whole new network was trained given the instances on which the prior networks had made mistakes at each iteration.

5 Stage Two: Merging Overlapping Detections and Arbitration

The examples in Fig. 3 showed that the raw output from a single network will contain a number of false detections. In this section, we present two strategies to improve the reli ability of the detector: merging overlapping detections from a single network and arbitrating among multiple networks.

5.1 Merging Overlapping Detections

The majority of faces are recognized at numerous neighboring locations or scales, but false detections are more common. As a result of this finding, a heuristic has been developed that can remove many erroneous detections. The number of detections within a defined neighborhood of a place may be tallied for each location and scale. If the number exceeds a certain threshold, the place is considered a face. The position of the detection result is defined by the centroid of

surrounding detections, resulting in many detections. If a place is successfully recognized as a face, any other detection locations that overlap it are likely to be mistakes and may be removed. We keep the site with the highest number of detections within a local neighborhood and reject places with less detection based on the aforementioned criteria for neighboring detections. This approach is referred to as "overlap removal" in the description of the trials. This heuristic fails in a small number of instances; nonetheless, one such scenario is depicted by the left two faces, where one partially occludes the other.ing collapsed. This heuristic will be referred to as "thresholding" in the experiments section.

The accompanying diagram shows how these two heuristics are implemented. Each detection is recorded in an image pyramid, dubbed the "output" pyramid, at a certain position and size. The amount of detections in a specific neighborhood of each site in the pyramid is then used to position each location in the pyramid. The detections are "spread out" as a result of this. The neighborhood normally extends an identical amount of pixels in both size and position, with detections only spread out in position. These values are given a threshold, and the centroids (in both location and scale) of all areas over the threshold are calculated. All detections that contribute to a centroid's location are merged into a single point. The centroids are then examined one by one, beginning with the ones that had the most detections within the specified neighbourhood. The output pyramid is deleted if any additional centroid locations indicate a face that overlaps with the current centroid. The final detection result is made up of all remaining centroid positions. Similar observations about the nature of the outputs were made in the face detection work described in [3], resulting in the development of heuristics similar to those described above.

5.2 Arbitration Among Multiple Networks

We can apply multiple networks and arbitrate between their outputs to produce the final decision to reduce the number of false positives even more. Each network is trained in a similar manner, but with random initial weights, random initial nonface images, and permutations of the order of presentation of the scenery images. The detection and false-positive rates of the individual networks will be quite close, as will be seen in the next section. However, because of different training conditions and because of self-selection of negative training examples, the networks will have different biases and will make different errors. As with the previous heuristics, the implementation of arbitration shows that each detection at a specific position and scale is recorded in an image pyramid. ANDing two pyramids is one way to combine them. This strategy signals a detection only if both networks detect a face at precisely the same scale and position. Due to the different biases of the individual networks, they will rarely agree on a false detection of a nonface. This allows ANDing to eliminate most false detections. Unfortunately, this heuristic can decrease the detection rate because a face detected by only one network will be thrown out. However, we will see later that individual networks can all detect roughly the same set of faces, so that the number of faces lost due to ANDing is small.

Similar strategies were also tested, such as ORing the outputs of two networks or voting among three networks. Each of these arbitration procedures can be used either before or after the heuristics of "thresholding" and "overlap removal." If we apply it afterwards, we combine the centroid positions rather than the actual detection sites, and we require them to be at a reasonable distance of one another rather than perfectly aligned.

Arbitration methods like ANDing, ORing, and voting may appear intuitively logical, but there may be other less evident heuristics that work better. We used a second neural network to arbitrate between different detection networks to test this idea. In the output pyramid of each individual network, the arbitration network analyses a small area surrounding a place of interest. We count the number of detections in a 3 * 3 pixel zone surrounding the site of interest at each of three scales for each pyramid, resulting in three values for each detector, which are sent to the arbitration network.

Location contains a face and to produce a negative output for locations without a face. As will be seen in the next section, using an arbitration network in this fashion produced results comparable to (and in some cases, slightly better than) those produced by the heuristics presented earlier.

6 Experiment Results

To assess the system, a variety of tests were carried out. We first examine which characteristics the neural network uses to recognise faces, then demonstrate the system's error rates across two big test sets.

6.1 Sensitivity Analysis

We used the approach of [2] to do a sensitivity study to see which part of the input picture the network utilizes to identify whether the input is a face. We gathered a positive test set of face pictures based on the training database, but with different randomized scales, translations, and rotations than the training database. The negative test set was created from a collection of negative samples gathered while other networks were being trained. 100 2* 2 pixel sub pictures were created from each of the 20 * 20 pixel input photos. We proceeded through the test set one by one, replacing each sub picture with random noise and testing the neural network. On the test set, the resultant root mean square error of the network indicates how significant that region of the picture is for the detection job. The networks rely on the eyes first, then the nose, and finally the mouth.

6.2 Testing

The system was put to the test on two big collections of pictures that were not the same as the training sets. Test Set 1 comprises of 130 images gathered at CMU, including photos from the Internet, scanned from photographs and newspaper pictures, and digitized from television broadcasts. It also includes 23 pictures that were used to test the system's accuracy. The pictures include 507 frontal faces in total, requiring the networks to analyze 83,099,211 20 *20 pixel windows. The pictures feature a range of diverse backgrounds and are valuable for determining the system's false-alarm rate. The FERET database [16], [17] is a subset of Test Set 2.

Each photograph features one face, a homogeneous background, and decent lighting (in most situations). The collection contains a wide range of faces photographed from various perspectives. As a result, these pictures are more useful for assessing the detector's angular sensitivity than for measuring the false-alarm rate.

Our face detection networks produce non-binary results. The neural network outputs real values ranging from 1 to -1, indicating if the input comprises a face. During training, a threshold value of zero is utilized to pick the negative instances (if the network outputs a value of greater than zero for any input from a scenery image, it is considered a mistake). Although this figure appears to be fair, we may alter the system's conservatism by adjusting it during testing. We assessed the detection and false-positive rates when the threshold was adjusted from 1 to -1 to examine the influence of this threshold value during testing. The false-detection rate is zero at a threshold of 1, yet no faces are recognised. As the threshold is lowered, the number of true detections rises, but the number of erroneous detections rises as well. This trade-off is depicted in the graph, which displays the detection rate vs the number of false positives when the threshold is adjusted for the two networks discussed before. We utilised a zero threshold value throughout testing since the zero threshold sites are near to the curves' "knees," as seen in the image.

On Test Set 1, Table 1 illustrates the performance of several detector variants. The number of faces overlooked (out of 507), the detection rate, the total number of erroneous detections, and the false-detection rate are all displayed in the four columns. The final rate is expressed in terms of the number of 20 * 20 pixel windows that must be inspected, which is around 3.3 times the total amount of pixels in a picture (taking into account all he levels in the input pyramid). We first evaluated four networks on their own, then looked at the effects of overlap removal and compressing numerous detections, and then used ANDing, ORing, voting, and neural networks to test arbitration.

Networks 3 and 4 are the same as Networks 1 and 2, except that during training, the negative example pictures were shown in a different order. Networks 1 and 2 provided the results for ANDing and ORing networks, while Networks 1, 2, and 3 provided the results for voting and network arbitration. The pictures from which the face examples were collected were used to train the neural network arbitrators. The network arbitrator or three distinct designs were utilized. Five

concealed units were utilised in the first. The second employed two five-unit hidden layers, each with complete connections between them and extra connections between the first hidden layer and the output. The last architecture was a simple perception, with no hidden units

The "thresholding" heuristic for merging detections, as previously mentioned, requires two parameters: the size of the neighborhood utilized in looking for neighbouring detections and the number of detections that must be detected in that area. These two factors are listed in parenthesis following the term "threshold" in the table. The ANDing, ORing, and voting arbitration techniques all contain a parameter that specifies how close two detections (or detection centroids) must be to be considered as identical.

The raw performance of the networks is shown in Systems 1 through 4. The networks in Systems 5 through 8 are the identical, but they contain thresholding and overlap removal stages, which dramatically reduce the amount of false detections at the cost of a minor drop in detection rate. The remaining systems all rely on multi-network arbitration. Arbitration lowers the false-positive rate even further and, in certain circumstances, somewhat enhances the detection rate. Note that the ratio of false detections to windows investigated for systems utilising arbitration is extremely low, ranging from one false detection 449,184/windows to one false detection 41,549,605, depending on the type of arbitration employed.

The detector can be adjusted to be more or less cautious, as shown in Systems 10, 11, and 12. System 10, which employs ANDing, produces a very low amount of false positives, with a detection rate of around 77.9

The detection and false-alarm rates of Systems 14, 15, and 16, which all employ neural network-based arbitration among three networks, are comparable to those of Systems 10 and 11. The accuracy of System 13, which relies on voting across three networks, is comparable to that of Systems 11 and 12. The following part will go through System 17. Based on the notional angle of the face with regard to the camera, we divided the pictures into three groups: frontal faces, faces at an angle of 15 degrees from the camera, and faces at an angle of 22.5 degrees. Within these groupings, the orientation of the face differs substantially. As shown in the table, the detection rate for systems arbitrating two networks varies between 97.8 percent and 100.0 percent for frontal and 15 degree faces, and between 91.5 percent and 97.4 percent for 22.5 degree faces. This is due to the fact that the training set consists primarily of frontal faces. It's worth noting that systems have a better detection rate for faces at a 15-degree angle than for frontal faces. The majority of persons with missing frontal faces are wearing glasses that reflect light into the camera. The detector hasn't been trained on photos like these, so it assumes the eyes are darker than the rest of the face. Thus the detection rate for such faces is lower.

We determined that both Systems 11 and 15 make acceptable trade-offs between the amount of false detections and the detection rate based on the data presented in Tables 1 and 2. System 11 is superior than System 15 since it is less complicated (it uses just two networks instead of four). In Test Set 1, System 11 correctly recognizes 86.2 percent of the faces, with an average of one incorrect detection every 3,613,009 20 * 20 pixel windows.

7 Improving the Results

In this part, we'll go through a few strategies for increasing the system's speed. The work discussed here is preliminary, and it is not meant to be a comprehensive examination of approaches for reducing execution time. The number of 20 *20 pixel windows that the neural networks must analyse is the most important component in the system's operating duration so far. On a 200 MHz R4400 SGI Indigo 2, applying two networks on a 320 * 240 pixel picture (246,766 windows) takes around 383 seconds. The arbitration stages have a low computational cost, requiring less than a second to merge the outputs of the two networks over all places in the picture.

Remember that the degree of position invariance in our system's pattern recognition component dictates the number of windows that must be processed. This was used to reduce the number of windows that needed to be processed in the associated task of licence plate detection. The goal was to make the neural network insensitive to translations of around 25

Face detection may be approached in the same way. A 20 * 20 face centred in a 20 * 20 window was trained into the original detector. Allowing the same 20 $\times$ 20 face to be off-center by up to five pixels in either direction allows us to make the detector more versatile. The window size has

been adjusted to 30 × 30 pixels to ensure that the network can still view the entire face. As a result, the centre of the face will be contained to a 10 * 10 pixel rectangle in the window's centre. The network has only one output, which indicates if a face is there or not. This detector can be dragged across the image in 10 pixel increments while still detecting all possible faces.

The scanning procedure is shown in the diagram, which displays the input picture pyramid and which of the 10 * 10 pixel sections are categorised as containing face centres. An image output architecture was also explored, which provided similar detection accuracy but required more processing. The same bootstrap approach as before was used to train the network. Before being sent to the network, the windows are preprocessed with histogram equalisation.

As can be observed in the graph, this network has a lot more false detections than the previous detectors. To increase accuracy, we regard each 30 * 30 detector detection as a candidate and validate it using the 20 * 20 detectors stated before. The verification network's 20 * 20 window must be searched over the 10 * 10 pixel region possibly holding the centre of the face since the candidate faces are not exactly positioned. The outputs of two verification networks are combined using a simple arbitration approach called ANDing. By initially scanning the picture for huge faces and avoiding processing places that overlap with any detections identified so far, the heuristic that faces seldom overlap may be leveraged to decrease computation. On a 200 MHz R4400 SGI Indigo 2, a typical 320 * 240 picture takes roughly 7.2 seconds to process with these changes. The redesigned system was applied to the test sets used in the previous section to see how these modifications affected the system's accuracy. As can be observed, this system has detection and false-alarm rates that are equivalent to System 10, which is the most cautious of the other systems. If you're analysing a large number of photos captured by a stationary camera, you can get even better results. By photographing the backdrop scene, one may detect which parts of the image have changed in a freshly collected image and examine only those parts.

8 Conclusion

In a collection of 130 test photos, our system can recognise between 77.9

Future work might go in a variety of areas. The present system's fundamental flaw is that it can only recognise upright faces staring at the camera. For each head orientation, several copies of the system could be trained, and the results may be integrated using arbitration procedures similar to those described here. Because profile views of faces have fewer stable characteristics and the input window contains more background pixels, preliminary work in this field suggests that recognising profile views of faces is more challenging than detecting frontal views. The same system has also been used to identify automobile tyres and human eyeballs, however further study is needed.

More effort is needed even in the realm of identifying frontal views of faces. When a picture sequence is available, temporal coherence can be used to draw attention to certain areas of the photos. As a face travels about, its position in one frame is a good predictor of where it will be in the next. To concentrate the detector's attention, standard tracking approaches as well as expectation-based methods can be used. Obtaining more positive examples for training or using more complex picture preprocessing and normalization techniques are two more ways to improve system performance.

The field of media technology is one application of this research. Every year, advances in technology make it easier and less expensive to store and retrieve visual data. However, the ability to automatically classify information material at a high level is severely constrained; this is a bottleneck that hinders media technology from realising its full potential. Users can ask systems using the above-mentioned detector to "show me the persons who appear in this movie" [18], [20], or "Which photos on the World Wide Web contain faces?" [6] and to have their questions automatically addressed.

9 References

1) S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Technical Report CMU-CS-94-163, Carnegie Mellon Univ., 1994

2) S. Baluja, "Expectation-Based Selective Attention," PhD thesis, Carnegie Mellon Univ. Computer Science Dept., Oct. 1996. Available as CS Technical Report CMU-CS-96-182.

3) G. Burel and C. Carel, "Detection and Localization of Faces on Digital Images," Pattern Recognition Letters, vol. 15, pp. 963-967, Oct. 1994

4) A.J. Colmenarez and T.S. Huang, "Face Detection With Information-Based Maximum Discrimination," Computer Vision and Pattern Recognition, pp. 782-787, 1997.

5) H. Drucker, R. Schapire, and P. Simard, "Boosting Performance in Neural Networks," Int'l J. Pattern Recognition and Artificial Intelligence, vol. 7, no. 4, pp. 705-719, 1993.

6) C. Frankel, M.J. Swain, and V. Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," Technical Report TR-96- 14, Univ. of Chicago, Aug. 1996.

7) V. Govindaraju, "Locating Human Faces in Photographs," Int'l J. Computer Vision, vol. 19, no. 2, pp. 129-146, 1996.

8) J. Hertz, A. Krogh, and R.G. Palmer, Introduction to the Theory of Neural Computation. Reading, Mass.: Addison-Wesley Publishing Company, 1991.

9) H.M. Hunke, "Locating and Tracking of Human Faces With Neural Networks," master's thesis, Univ. of Karlsruhe, 1994.

10) Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, vol. 1, pp. 541-551, 1989.

11) T.K. Leung, M.C. Burl, and P. Perona, "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching," Proc. Fifth Int'l. Computer Vision, pp. 637-644, Cambridge, Mass., June 1995.

12) S.H. Lin, S.Y. Kung, and L.J. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network," IEEE Trans. Neural Networks, Special Issue on Artificial Neural Networks and Pattern Recoognition, vol. 8, no. 1, Jan. 1997.

13) B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Detection," Proc. Fifth Int'l Conf. Computer Vision, pp. 786-793, Cambridge, Mass., June 1995.

14) E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," Computer Vision and Pattern Recognition, pp. 130-136, 1997.

15) A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," Computer Vision and Pattern Recognition, pp. 84-91, 1994.

16) P.J. Phillips, H. Moon, P. Rauss, and S.A. Rizvi, "The FERET Evaluation Methodology for Face-Recognition Algorithms," Computer Vision and Pattern Recognition, pp. 137-143, 1997.

17) P.J. Phillips, P.J. Rauss, and S.Z. Der, "FERET (Face Recognition Technology) Recognition Algorithm Development and Test Results," Technical Report ARL-TR-995, Army Research Lab., Oct. 1996.

18) S. Satoh and T. Kanade, "Name-It: Association of Face and Name in Video," Computer Vision and Pattern Recognition, pp. 368-373, 1997.

19) P. Sinha, "Object Recognition Via Image Invariants: A Case Study," Investigative Ophthalmology and Visual Science, vol. 35, no. 4, Mar. 1994.

20) M.A. Smith and T. Kanade, "Video Skimming and Characterization Through the Combination of Image and Language Understanding Techniques," Computer Vision and Pattern Recognition, pp. 775-781, 1997.

21) K.-K. Sung, "Learning and Example Selection for Object and Pattern Detection," PhD thesis, MIT AI Lab, Jan. 1996. Available as AI Technical Report 1572.

22) T. Umezaki, personal communication, 1995.

23) R. Vaillant, C. Monricq, and U. Le Cun, "Original Approach for the Localization of Objects in Images," IEE Proc. Vision, Image, and Signal Processing, vol. 141, no. 4, Aug. 1994.

24) A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," Readings in Speech Recognition, pp. 393-404, 1989.

25) G. Yang and T.S. Huang, "Human Face Detection in a Complex Background," Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994.

26) K.C. Yow and R. Cipolla, "Feature-Based Human Face Detection," Technical Report CUED/F-INFENG/TR 249, Dept. of Eng., Univ. of Cambridge, England, 1996.