December 2005

A Fast Neural-Based Eye Detection System

Fok Hing Chi Tivive
University of Wollongong, tivive@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

A Fast Neural-Based Eye Detection System

Abstract

This paper presents a fast eye detection system which is based on an artificial neural network known as the shunting inhibitory convolutional neural network, or SICoNNet for short. With its two-dimensional network architecture and the use of convolution operators, the eye detection system processes an entire input image and generates the location map of the detected eyes at the output. The network consists of 479 trainable parameters which are adapted by a modified Levenberg-Marquardt training algorithm in conjunction with a bootstrap procedure. Tested on 180 real images, with 186 faces, the accuracy of the eye detector reaches 96.8% with only 38 false detections.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Tivive, FHC & Bouzerdoum, A, A Fast Neural-Based Eye Detection System, Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2005), 13-16 December 2005, 641-644.

A FAST NEURAL-BASED EYE DETECTION SYSTEM

Fok Hing Chi Tivive and Abdesselam Bouzerdoum, Senior Member, IEEE

School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Northfields Avenue, Wollongong, NSW 2522, AUSTRALIA.
E-mail:[fhct243@uow.edu.au], [a.bouzerdoum@elec.uow.edu.au]

ABSTRACT

This paper presents a fast eye detection system which is based on an artificial neural network known as the *shunting inhibitory convolutional neural network*, or SICoNNet for short. With its two-dimensional network architecture and the use of convolution operators, the eye detection system processes an entire input image and generates the location map of the detected eyes at the output. The network consists of 479 trainable parameters which are adapted by a modified Levenberg-Marquardt training algorithm in conjunction with a bootstrap procedure. Tested on 180 real images, with 186 faces, the accuracy of the eye detector reaches 96.8% with only 38 false detections.

1. INTRODUCTION

In recent years, biometric recognition, such as iris and face recognition, have proven to be challenging research topics for the computer vision community. These physiological characteristics are becoming acceptable distinctive personal traits that have the potential to be applied in surveillance systems for identifying and verifying individuals. The face is one of the biometric characteristics that humans use in everyday life for personal recognition, and has been used by successful niche companies to develop biometric identification systems. Iris is another biometric that exhibits both permanence and individuality, and has been applied as an unobtrusive personal identification in security systems. One of the stepping stones of these recognition systems is the detection and localization of the human eyes.

Many studies dealing with the detection and verification of human eyes have been reported. They can be categorized into three groups: image-based approach, model-based approach and neural-based approach. In the image-based approach, color, texture, shape and motion have been used as important cues for eye detection. In [1], color information is used to detect skin regions and locate candidate eye patterns within or nearby the skin regions. However, this technique can only be applied to quasi-frontal and close-up facial images. Based on the physiological properties of the eye, some researchers [2, 3] have used infra-red illumination to detect the eyes. The approach is to focus an infra-red beam onto the eye. The cornea reflects back the infra-red beam causing the *red-eye* effect which is often seen in flash photographs. This phenomenon makes the pupil of the eye brighter in a gray-scale image, thereby facilitating the detection of the eyes. However, there are many objects in the image that exhibit similar reflectance properties, and hence cannot be distinguished from the eyes. Therefore, the success of these systems is very much dependent on the special illumination setup, the synchronization scheme, and other additional information about the eyes.

In the model-based approach, Yuille *et al.* [4] used template matching to detect the eye regions. The eye template is built from a circle, two intersecting parabolic curves and two points in the center of the white of the eye. The template is matched with the input image by minimizing an energy function. Later, Xie *et al.* [5] improved further the eye deformable template by including extra terms in the energy function used to determine the parameters of the template. Often, these template matching techniques do not produce accurate results, and they are quite sensitive to the initial parameters of the eye template [6]. Besides, they are time-consuming operations.

On the other hand, the neural-based approach offers the ability to learn directly from real input patterns and generates complex decision boundaries. In a two dimensional (2-D) space, convolutional neural networks (CoNNs) have been used successfully to solve 2-D pattern recognition tasks. Furthermore, they are renowned for having in-built tolerances for shift, rotation and distortion. Initially, they were developed to model the mammalian visual system, e.g., the *Cognitron* [7]. Later on, the emphasis of convolutional neural networks has been shifted to build powerful visual pattern recognition systems, such as character recognition [8] and face detection [9], to name a few.

In this paper, we have developed an eye detector based on a convolutional neural network. This CoNN has a generic network architecture, in which the feature extraction neurons are based on the bio-physical mechanism of shunting

0-7803-9266-3/05/$20.00 ©2005 IEEE.

inhibition. As each layer of the network acts as a convolution filter followed by a down-sampling operation, the eye detection system can process an entire input image and generate an output location map which is four times smaller than the original input image. With this detection procedure, the detector can be operated as a real-time system.

The remainder of the paper is organized as follows. The next section describes the network structure that has been adopted as an eye classifier. Section 3 explains the training technique and the eye detection procedure. Section 4 presents the experimental results and performance analysis. Finally, concluding remarks are presented in Section 5.

2. NETWORK ARCHITECTURE

The main module of the eye detection system is the neural-based eye classifier derived from the convolutional neural network architecture presented in [10]. The network consists of three layers: the first two layers are hidden layers with planes of neurons known as *feature maps* and the last layer consists of a single neuron, the output neuron. The first layer has two feature maps and the second layer has four feature maps. The connection scheme between these feature maps is similar to a binary tree, that is each feature map is connected to two feature maps in the following layer. Each feature map is made up of a lattice of shunting inhibitory neurons, which receive inputs from a local neighborhood, called the *receptive field* in the input image. The input layer is a 2-D image of size 32×32.

The response of a shunting inhibitory neuron can be mathematically described by

$$z_j = \frac{g\Big(\sum_i w_{ji} I_i + b_j\Big)}{a_j + f\Big(\sum_i c_{ji} I_i + d_j\Big)}, \quad \text{for } i = 1, \ldots, N \qquad (1)$$

where z_j is the activity of the j^{th} neuron, I_i's are external inputs, a_j is the passive decay rate, w_{ji} and c_{ji} are the connection weights from the i^{th} neuron to the j^{th} neuron, b_j and d_j are bias terms, N is the number of inputs within the receptive field, and f and g are activation functions. In the first layer, g and f are chosen to be the hyperbolic tangent and exponential functions, respectively, whereas in the second layer, g is set to the logarithmic sigmoid function. We should note that even though the input is a 2-D pattern, in (1) the input signal is a column vector; this can be achieved by concatenating the columns of the 2-D input.

All neurons in a feature map have the same set of weights (weight sharing) and the same bias parameters including the passive decay rate term. Within each layer, a sub-sampling operation is performed by shifting the centers of receptive fields of adjacent neurons by two positions, horizontally and vertically. This decreases the size of the feature maps by one quarter in successive layers. In the first and second layer of the network, the respective field sizes are 7×7 and 5×5. The inputs to the output layer are the local averages of 2×2 non-overlapping regions from all feature maps in the second layer; that is, each 2×2 region in a feature map provides one input signal to the output layer. The weighted sum of these locally averaged signals are passed through a linear activation function to generate the output of the neuron in the last layer of the network. Thus, the response of the output unit is given by

$$y = h\Big(\sum_v w_v z_v + b\Big), \quad \text{for } v = 1, \ldots, N_F \qquad (2)$$

where h is the output activation function, w_v's are the connection weights, z_v's are the inputs to the neuron, N_F is the number of inputs to the output layer, and b is the bias term.

3. EYE DETECTION SYSTEM

The network parameters are adapted by a batch training algorithm proposed by Ampazis and Perantonis [11]. It is a modified Levenberg-Marquardt (LM) training algorithm with an adaptive momentum term. Eye patterns cropped from Web images are used as training data. Some examples of the eye patterns are shown in Fig. 1. The entire eye database contains images with different eye apertures and orientations collected from people of different races, ages and gender. A pre-processing technique, such as range normalization, is applied on the training set so as to prevent any neurons of the network from falling into the saturation regions of the activation functions; hence the input patterns are linearly scaled or mapped to the range $[-1, 1]$. To classify the input images, the target values of the network are set to 1 for an eye and -1 for a non-eye pattern.

Fig. 1. Samples of eye (top) and non-eye (bottom) patterns from the training and test sets.

3.1. Network Training

The training methodology used in this experiment is a modified version of the bootstrap training technique proposed by Sung *et al.* [12], in conjunction with the modified LM training algorithm. The training strategy can be explained as follows. Initially, a training set of 500 patterns with equivalent number of eye and non-eye patterns is generated. Another disjoint set, the *cross-validation set*, is also generated for selecting the network with the minimum validation error. The network is trained for a certain number of epochs, and the trained network with the lowest validation error is used for

the next bootstrap session. A set of images, containing people with the eye regions removed, are used for collecting non-eye patterns. The trained network is applied on these images, and windows which have network responses greater than zero are considered as false alarms. In each bootstrap session, 500 of these false alarms together with the same number of eye patterns are added to the training set. Initially, distinct blocks of the same size as the input plane of the network are extracted from the images. Subsequently, after scanning a certain number of images, a sliding window with steps of four pixels is applied to scan the input image. Moreover, each image is sub-sampled to generate a series of multi-resolution images which are filtered by the trained network for non-eye patterns. The whole training process is stopped when the number of patterns in the training set reaches 15000.

3.2. Eye Localization Procedure

Since the eyes can appear at different sizes, the input image is sub-sampled at different resolutions with a scaling factor of 1.2 to form a pyramid of images. Each scaled image from the pyramid of images is processed by the network to generate an image of network responses. A common strategy to construct this output image from a scaled image is to extract individual input window at every position on the scaled image and feed it to the network to compute a network response. However, it is a time-consuming operation. As the receptive fields in a CoNN behave as convolutional kernels, the entire scaled image can be passed to the network, and at each hidden layer the image is convolved with the receptive fields and down-sampled by a factor of two in both dimensions. This results in an output image which is $1/16$th the size of the scaled image. This convolutional computation is faster than the scanning window operation, as it reduces the computation redundancy between two sliding windows. In the output image, the network responses that are greater than a threshold are considered as eye candidates and their locations are mapped back to a map which has the same size as the original input image.

During the detection phase, a certain number of background windows are often misclassified as eye candidates and have high network responses. Moreover, overlapping detections usually occur around the true eyes. To reduce these type of errors, the following post-processing steps are performed. The eye candidate is mirrored on the Y-axis and passed back to the network. The average of both network responses is taken as the final score of the positive detection. If the average network response is less than a threshold it is set to zero in the map. This double verification strategy reduces the number of false detections and increases the stability of the eye score. To group the overlapping detections into a representative eye candidate, a similar grouping technique explained in [9] has been used. In each map, all the eye candidates are gathered into a cluster which has a representative eye whose position is taken as the centroid of all the eye candidates and its confidence score is computed as the product of the highest score of the eye candidate and the total number of eye candidates within the cluster. If the cluster has less than three eye candidates, the representative eye candidate is removed from the map. All the maps are collapsed into a final map which is processed again by the grouping method. Fine searches in location and size of the eye are performed. The absolute position of the eye is sought in a region of eight pixels around the center of the representative eye. The location of all positive detections within the search grid are averaged to give the final position of the eye location. The representative eye is tested at nine scales of its detected size, ranging from 0.5 to 1.5, and the sizes of all positive detections are averaged to compute the final size of the eye. Furthermore, to verify the representative eye candidates, the number of positive detection is counted, and if it is greater than two, the representative eye candidate is accepted. Finally, the remaining representative eyes are passed to the network to verify their confidence scores.

4. RESULTS AND PERFORMANCE ANALYSIS

In pattern recognition and image processing, histogram equalization is commonly used to improve the contrast of an image. However, this technique is dependent on the number of image pixels. In other words, its output from a training pattern image is different when the technique is applied on an entire image. Therefore, to determine whether this technique improves the classification performance of the eye classifier, two networks of the same size were trained. The first network used training patterns that were histogram equalized and range normalized, and the second network used training patterns that were only range normalized. The training process was terminated at 200 epochs, and the network with the lowest validation error was selected. The performance evaluation was done on a disjoint test set that consists of 10000 eye patterns and 6 millions non-eye patterns. Figure 2 displays the *receiver operating characteristic* (ROC) curves of these networks. It shows that without the use of a histogram equalization method, the eye classifier achieves a 99% correct classification rate at 1% false detection rate. On the other hand, when using both preprocessing techniques, the classification performance of the eye classifier is reduced to 96%. Therefore, the input images to the network are only range normalized in subsequent tests. The eye detection system was evaluated on a set of 180 real images collected from the Web. These images have different sizes, and consist of people of different ages and gender, taken under different illumination conditions. To compute the system performance, the number of eyes

that have been correctly detected was counted. False detections and false dismissals were also recorded for statistical analysis. Table 1 summarizes the detection results of the eye detector when using either a double verification strategy (DVS) or a normal network evaluation (NNE) across the post-processing stage. In the NNE, the input pattern is processed by the network without mirroring. The thresholds used to build the pyramid of network response images and in the post-processing steps including DVS and NNE were zero.

Based on this test set, the neural-based eye detector can detect and localize human eyes at an accuracy of 96.8% with 38 false detections. Those detected eyes vary in size and orientation as shown in Fig 3. This demonstrates that the detector is quite tolerant to certain affine transformation. With a double verification strategy, there are a reduction of 2 false detections and 5 dismissals when both strategies used the same threshold. Lowering the threshold to -0.1 for NNE so that the correct detection rate is similar to DVS, it is clearly shown that DVS strategy has less false alarms in comparison to NNE, i.e., a reduction of 27 false detections. In comparison to the neural-based face detector proposed by Garcia and Delakis [9] which has 951 trainable parameters, the proposed network has fewer trainable parameters (479 trainable weights), and the main difference is that the eye detection system uses shunting inhibitory neurons as feature detectors, instead of perceptron neurons.

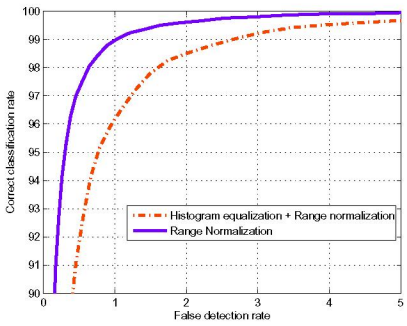

Fig. 2. ROC curves based on networks that used range normalization or histogram equalization + range normalization as pre-processing techniques.

5. CONCLUSION

In this paper, we have demonstrated that a convolutional neural network can be trained as an eye classifier with a correct classification rate of 99%, tested on segmented patterns. Based on this classifier, a fast eye detection system was developed. The system processes the entire input image and locates the eye patterns within the image. The detection accuracy of the system was around 97%. Furthermore, it has been shown that using a double verification strategy across the post-processing stage, the number of false detections has been reduced.

Table 1. Detection results of the eye detector.

Verification process	Correct Detection rate (%)	F. Detections	F. Dismissals
DVS (Tr = 0)	96.8	38	12
NNE (Tr = 0)	95.4	40	17
NNE (Tr = −0.1)	96.5	65	13

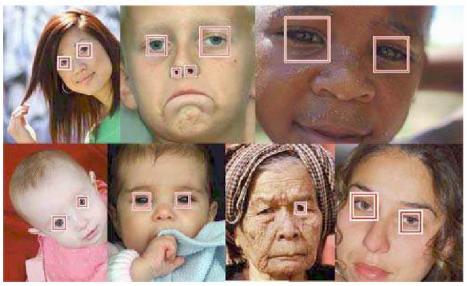

Fig. 3. Example of detected eye images.

6. REFERENCES

[1] R. T. Kumar, S. K. Raja, and A. G. Ramakrishnan, "Eye detection using color cues and projection functions," in *Proc. 2002 Int. Conf. on Image Processing*, 2002, vol. 3, pp. III–337–III–340.

[2] A. Haro, F. Myron, and E. Irfan, "Detecting and tracking eye by using their physiological properties, dynamics and appearance," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2000, vol. 1, pp. 163–168.

[3] K. Nguyen, C. Wagner, D. Koons, and M. Flickner, "Differences in the infrared bright pupil response of human eyes," in *Proc. of Eye Tracking Res. and Application Symposium*, ACM, New York, 2002, pp. 133–138.

[4] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[5] X. Xie, R. Sudhakar, and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, vol. 27, no. 6, pp. 791–799, 1994.

[6] H. Tan, Y. J. Zhang, and R. Li, "Robust eye extraction using deformable template and feature tracking ability," in *Proc. of the Joint Conf. of the Fourth Int. Conf. on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conf. on Multimedia*, 2003, vol. 3, no. 3, pp. 1747 – 1751.

[7] K. Fukushima, "Cognitron: A self organizing multilayered neural network," *Biological Cybernetics*, pp. 121–136, 1975.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 26, no. 11, pp. 1408–1423, 2004.

[10] F. H. C. Tivive and A. Bouzerdoum, "Efficient training algorithms for a class of shunting inhibitory convolutional neural networks," *IEEE Trans. on Neural Networks*, vol. 16, no. 3, pp. 541– 556, 2005.

[11] N. Ampazis and S. J. Perantonis, "Two highly efficient second-order algorithms for training feedforward networks," *IEEE Trans. on Neural Networks*, vol. 13, no. 5, pp. 1064–1074, 2002.

[12] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. on Pattern Recognition and Machine Intell.*, vol. 20, no. 1, pp. 31–59, 1998.