2005

An eye feature detector based on convolutional neural network

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

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

An eye feature detector based on convolutional neural network

Abstract

One of the main problems when developing an eye detection and tracking system is to build a robust eye classifier that can detect the true eye patterns in complex scenes. This classi fication task is very challenging as the eye can appear in different locations with varying orientations and scales. Furthermore, the eye pattern varies intrinsically between ethnic groups, and with age and gender of a person. To cope better with these variations, we propose to use a bio-inspired convolutional neural network, based on the mechanism of shunting inhibition, for the detection of eye patterns in unconstrained environments. A learning algorithm is developed for the proposed neural network. Experimental results show that such network has the builtin invariant knowledge and the discriminatory power to classify input regions into eye and non-eye patterns. A classification rate of 99% is achieved by a three layer network with input size of 32 x 32 pixels.

Disciplines

Physical Sciences and Mathematics

Publication Details

F. Tivive & A. Bouzerdoum, "An eye feature detector based on convolutional neural network," in The Eight International Symposium on Signal Processing and Its Applications, 2005, pp. 90-93.

AN EYE FEATURE DETECTOR BASED ON CONVOLUTIONAL NEURAL NETWORK

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-mails: fhct243@uow.edu.au, a.bouzerdoum@elec.uow.edu.au

ABSTRACT

One of the main problems when developing an eye detection and tracking system is to build a robust eye classifier that can detect the true eye patterns in complex scenes. This classification task is very challenging as the eye can appear in different locations with varying orientations and scales. Furthermore, the eye pattern varies intrinsically between ethnic groups, and with age and gender of a person. To cope better with these variations, we propose to use a bio-inspired convolutional neural network, based on the mechanism of shunting inhibition, for the detection of eye patterns in unconstrained environments. A learning algorithm is developed for the proposed neural network. Experimental results show that such network has the built-in invariant knowledge and the discriminatory power to classify input regions into eye and non-eye patterns. A classification rate of 99% is achieved by a three layer network with input size of 32×32 pixels.

1. INTRODUCTION

As we step into a new era of intelligent man-machine interactions, human facial features are attracting considerable interest from the research community. Features such as mouse, nose or eyes provide strong cues for recognition and tracking of human faces in complex scenes. The human eye, for example, is considered an important salient feature that provides crucial information for fatigue analysis, visual interpretation, detection and recognition of human faces, object-based coding, etc..

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 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, 4] 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 are very much dependent on the special illumination setup, the synchronization scheme, and other additional information about the eyes.

In model-based approach, Yuille *et al.* [5] 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 to the input image by minimizing an energy function. Later, Xie *et al.* [6] 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 [7]. In addition, they are time-consuming operations.

Artificial neural networks have been applied to pattern recognition problems where traditional methodologies have failed or are very complicated to build. They offer an ability to perform tasks outside the scope of traditional processors such as parallel computing and fault-tolerance. Zhang *et al.* [8] developed a hybrid neural network to detect eye candidates from an input facial image. The network consists of radial basis functions as processing units in the hidden layer and sigmoid units at the output layer to classify each scanned window into eye and non-eye pattern. To reduce the number of false eye candidates, they developed a set of rules based on the geometric knowledge of the eye and its location within the facial region. Furthermore, to achieve scale and rotation invariance, they applied the c-means clustering algorithm to determine the centers of each hidden unit.

In this paper, we propose to use a convolutional neural network to classify image regions into eye and non-eye patterns, irrespective of the pattern orientation. This type of neural networks has been renowned of having some built-in tolerance for shift, translation, and distortion. The key characteristics of this type of networks are the connections from one layer to another, which are done through a set of biologically motivated receptive fields that are shared among processing units, and the arrangement of

0-7803-9243-4/05/$20.00 ©2005 IEEE

processing units that has some degree of similarity with biological vision systems. Moreover, the feature extraction stage is integrated with the classification stage, and both are generated by the learning process. The rest of this paper is organized as follows. The next section gives a description of the proposed neural network structure to be used for eye detection. In addition, the training technique is briefly explained in this section. Section 3 presents the experimental results and analyzes the network performance. Section 4 presents concluding remarks.

2. NEURAL NETWORK MODEL

Several convolutional neural network architectures have been proposed in the past, and most of them are based on three structural concepts: local receptive fields, weight sharing and sub-sampling. However, these networks differ markedly in their implementations; most importantly, they are specifically tailored for given tasks, which limit their use and applicability to other tasks. In this paper, we adopt a generic convolutional neural network (CoNN) architecture, in which the feature extraction neurons are based on the bio-physical mechanism of shunting inhibition. Although this CoNN architecture is herein applied to eye detection, it can easily be adapted for other image recognition tasks. The architecture consists of two hidden layers, an input layer and an output layer. The input layer receives a two-dimensional (2-D) input of arbitrary size. The first hidden layer contains two planes of processing units known as *feature maps*, each of which branches out to two feature maps in the succeeding layer. Consequently, the second hidden layer contains four feature maps. The connection between the feature maps is similar to a binary tree (see [9] for details). Each feature map is made up of a lattice of shunting neurons. These neurons receive input signals from small local regions of the input image, called *receptive field*. The activation of a shunting inhibitory neuron can be mathematically described by

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

where z_j is the activity of the j^{th} neuron, I_i's are the 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 constant biases, N is the size of 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 the neurons in the feature map share the same set of weights (weight sharing) and the same bias parameters including the passive decay rate. This process constrains each unit in the feature map to perform the same operation on different parts of the image. Consequently, the same elementary visual feature is extracted from different positions of the input image. Other feature maps in the layer perform the same operation with different sets of weights to extract different types of local features. Within each layer, a sub-sampling operation is performed by shifting the centers of receptive fields of neighboring units by two positions, horizontally and vertically. This decreases the size of the feature maps by one quarter in successive layers, and introduces some degree of shift and distortion invariance into the network. The same receptive field size of 5×5 is used to connect one layer to the next layer throughout the entire network architecture. The output layer of the network consists of one perceptron neuron. The inputs to the output layer are the local average 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 an appropriate activation function to generate the output. Thus, the response of an output unit is given by

$$y = h\Big(\sum_v w_v z_v + b\Big), \qquad (2)$$

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

3. TRAINING AND EVALUATION

3.1. Training Methodology

In [9], a series of training algorithms, ranging from first-order gradient methods to Quasi-Newton, have been developed for the proposed CoNN. In the experiments presented herein, we have adopted the algorithm proposed by [10], and the final formula to compute the weight update $\Delta\vec{W}(k)$ is defined as

$$\Delta\vec{W}(k) = -\frac{\lambda_1}{2\lambda_2}\left[\mathbf{G}(k)\right]^{-1}\vec{g}(k) + \frac{1}{2\lambda_2}\Delta\vec{W}(k-1), \qquad (3)$$

The matrix $\mathbf{G}(k) = \mathbf{J}^T(k)\mathbf{J}(k) + \mu(k)\mathbf{I}$, where $\mathbf{J}(k)$ is the Jacobian matrix at the k^{th} iteration, $\mu(k)$ is a regularization parameter, and $\mathbf{I}$ is the identity matrix. The gradient vector $\vec{g}(k) = \mathbf{J}^T(k)\vec{e}(k)$, where $\vec{e}(k)$ is the error vector at the k^{th} iteration. The constants λ_1 and λ_2 are given by

$$\lambda_1 = \frac{-2\lambda_2\alpha + I_{GF}}{I_{GG}}; \quad \lambda_2 = \frac{1}{2}\left[\frac{I_{FF}I_{GG} - I_{GF}^2}{I_{GG}\beta^2 - \alpha^2}\right]^{\frac{1}{2}};$$

where

$$I_{GF} = \vec{g}^T(k)\Delta\vec{W}(k-1), \quad \alpha = -\beta\sqrt{I_{GG} - \frac{I_{GF}^2}{I_{FF}}},$$
$$I_{GG} = \vec{g}^T(k)\left[\mathbf{G}(k)\right]^{-1}\vec{g}(k), \quad \beta = 2\sqrt{I_{GG}},$$
$$I_{FF} = \Delta\vec{W}^T(k-1)\mathbf{G}(k)\Delta\vec{W}(k-1).$$

The parameter $\mu(k)$ is adapted based on the first Wolfe condition:

$$E(k+1) \leq E(k) + 0.1\vec{g}^T(k)\Delta\vec{W}(k), \qquad (4)$$

where $E(k)$ is the error function. If the condition holds, $\mu(k)$ is decreased by a factor of ten; otherwise it is increased by the same factor until there is a reduction in the error function. The Jacobian matrix is computed using a modified error-backpropagation rule similar to the one proposed by Hagan [11]. To terminate the training process, an early stopping procedure is used. This procedure trains an initialized network for 100 iterations, and at each iteration, the network is tested on a separate validation data set. Once the training process has terminated, the network with the lowest validation error is selected.

Before training commences, the weights of the receptive fields are initialized with random values using a uniform distribution between $-1/t$ and $1/t$, where t is the width of the receptive fields. The bias parameters b and d of the neurons in the feature maps are initialized similarly with $t = 1$. Moreover, the passive decay rate parameter a is initialized in the range $[0, 1]$, then it is constrained to satisfy the following condition:

$$a_j + f\Big(\sum_i c_{ji} I_i + d_j\Big) \geq 0.1. \quad (5)$$

This condition is applied to avoid division by zero in (1), and is maintained throughout the training process.

3.2. Training Data

Before the network can be used as an eye detector, it must be trained on eye and non-eye patterns. To this end, a training set is prepared, which contained eye and non-eye patterns, along with the corresponding desired outputs. The training process consists of adapting the network parameters so as to reduce the disparity between the network output and the given target value. The eye patterns were obtained by cropping square windows covering the eye envelope from images collected from the Web. The non-eye patterns, on the other hand, were obtained from images of natural scenes with no human faces. Some examples of the eye and non-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, with varying illumination conditions. To prevent any processing units of the network

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

from falling into the saturation regions of the activation functions, the input patterns are pre-processed by linearly scaling the pixel values of the gray scale image into the range $[-1, 1]$. The target values of the network are 1 for an eye and -1 for a non-eye pattern. Three training sets were prepared to train three network architectures having the following input size: 16×16, 24×24, 32×32. Each training set consisted of 8000 samples, with equal number of eye and non-eye patterns. A test set of 3000 eye patterns and 10000 non-eye patterns was used to analyze the performances of the trained networks. Furthermore, three networks of the same structure, but different initial weights, were generated for each input size. The average performances of the three networks are recorded in Table 1 for different input sizes.

4. RESULTS AND PERFORMANCE ANALYSIS

Since the network generates a response between $[-1, 1]$, a threshold is required to set the boundary between the eye and non-eye classes. The threshold value is empirically chosen so that the total classification error is at a minimum. Table 1 presents the average classification rates of the three networks on the test set. These results show that the proposed convolutional neural network architecture has the capability to detect eye patterns at different orientations and scales with very high classification accuracy. All trained networks achieve over 97% correct classification rate of eye patterns. From the *receiver operating characteristic* (ROC) curves of Fig. 2, the convolutional neural network achieves 99% correct detection rate at 1% false detection rate with input size of 32×32 pixels. However, when reducing the input size to a 16×16 pixels, the classification accuracy of the network drops significantly. One possible reason for such reduction in performance is the fact that the eye shape resembles two horizontal lines at low resolution, features which often appear in natural scenes. For this reason, many neural-based detectors [12, 13, 14] use a much larger input size. The first two

Table 1. The average generalization performances of the networks of different input size.

Retina size	Eye pattern (%)	Non-eye pattern (%)
16×16	97.3	96.7
24×24	98.0	98.7
32×32	99.0	99.0

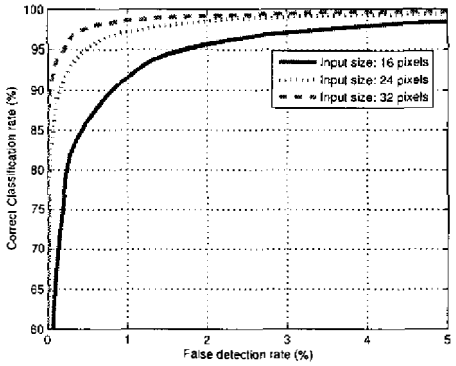

Fig. 2. The averaged ROC curves of the networks based on different input size.

layers of the network work as feature detectors by train-

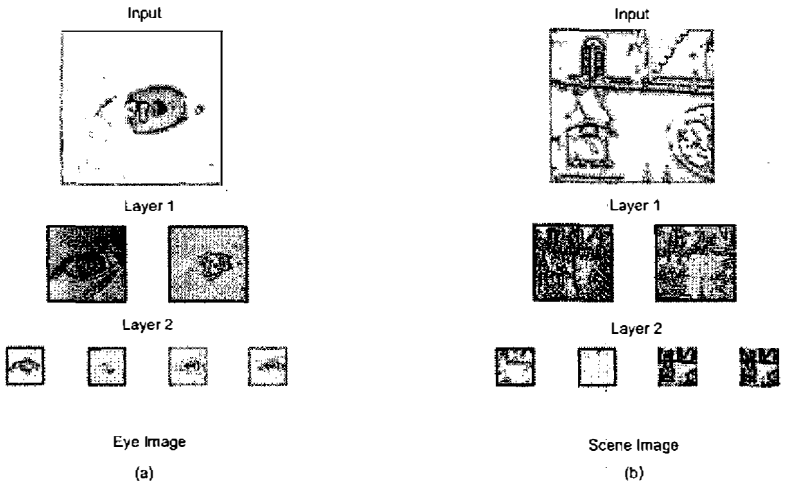

Fig. 3. The output images generated at the feature maps of the network.

ing the weights of the shunting inhibitory units as non-linear adaptive filters. To have an insight into what the processing units have learnt, two images of size 17×16 were processed by the network, and the output of each feature map is displayed in Fig. 3. The figure shows that the processing units in the feature maps behave like some kind of edge or feature detectors; they extract the salient features of the input image. For example, in Fig. 3(a), the eye is represented by parabolic curves which are different from those extracted features in Fig. 3(b). These features are further processed by the output unit to classify the input image into eye and non-eye patterns.

5. CONCLUSION

In this paper we have developed an eye detector which achieves a certain degree of tolerance to rotation and translation. It is based on a convolutional neural network which has a simple architecture and a systematic connection scheme. Experimental results show that such neural-approach can achieve a correct eye classification rate of 99%. A further investigation of the outputs of the feature maps revealed that the shunting inhibitory units behave as some kind of non-linear feature detectors.

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 Conf. on Computer Vision an Pattern Recognition*, 2000, vol. 1, pp. 163-168.

[3] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, pp. 357–377, 2002.

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

[5] 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.

[6] 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.

[7] 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.

[8] D. Zhang, H. Peng, J. Zhou, and S. K. Pal, "A novel face recognition using hybrid neural and dual eigenspaces methods," *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 32, no. 6, pp. 787–793, 2002.

[9] 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.

[10] 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.

[11] M. T. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Trans. on Neural Networks*, vol. 5, pp. 989–993, 1994.

[12] H. A Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[13] B. Fasel, "Robust face analysis using convolutional neural networks," in *Proc. of the Sixteenth Int. Conf. on Pattern Recognition*, 2002, vol. 2, pp. 11–15.

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