Convolutional Neural Networks for Eye Detection in Remote Gaze Estimation Systems

Jerry, Chi Ling Lam and Moshe Eizenman

***Abstract*—An eye detection algorithm based on Convolutional Neural Networks (CNN) architecture was developed. The algorithm was designed to detect eyes in video images from a remote gaze estimation system that is part of a gaze-controlled human-computer interface. The CNN for eye detection has two stages of convolutional and sub-sampling layers followed by a fully connected feed forward neural network with a total of 1227 trainable parameters. Experiments with 3 subjects showed that for the full range of expected head movements, the CNN achieved a detection rate of 100%, for images with fully opened eyes, and a false alarm rate of 2.65 X 10^{-4} %. The CNN failed to detect eyes that were either partially or completely covered by the eyelids. The CNN for eye detection did not require pre-processing or normalization and was shown to be robust to changes in scale, rotation and illumination of the eyes.**

***Index Terms*—Convolutional Neural Networks, Remote Gaze Estimation, Eye Detection, Image Processing**

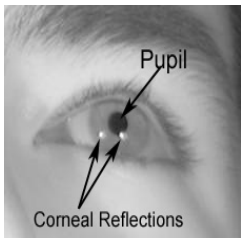
I. Introduction

The point-of-gaze (POG) is the point within the visual field that is imaged on the fovea; the highest acuity region of the retina. Systems that estimate the POG are used in a variety of applications such as studies of mood disorders [1] and driver behavior [2]-[4], pilot training [5], ergonomics [6], human-computer interfaces [7] and assistive devices for motor-disabled persons [8], [9].

Most modern remote gaze estimation systems are based on the analysis of specific eye features extracted from video images [10]. These specific eye features are the pupil center and a set of corneal reflections. The corneal reflections are virtual images of the light sources that illuminate the eyes. When the field of view of the eye tracker's video camera is limited, it is relatively easy to determine the eye features (Figure 1a). But, when the field of view of the camera is increased to allow for a larger range of head movements, the detection of eye features becomes much more difficult (Figure 1b). With a larger field of view, the number of false detections increases and the performance of the remote gaze estimation system deteriorate. It is therefore essential to develop algorithms that can detect reliably regions in the image that contain the eyes.

Jerry, Chi Ling Lam is a M.A.Sc. student with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G9, Canada (e-mail: jerry.lam@utoronto.ca).

*Moshe Eizenman is with the Departments of Electrical and Computer Engineering and Ophthalmology, and the Institute of Biomaterials and Biomedical Engineering, University of Toronto, 4 Taddle Creek Road, Toronto, ON M5S 3G9, Canada (phone: 1-416-978-5523; fax: 1-416-978-4317; e-mail: eizenm@ecf.utoronto.ca).



a)

b)

Figure 1: Eye Tracker's Images a) small field of view, b) larger field of view

Eye detection can be broadly classified into feature-based approaches and pattern-based approaches. Feature-based approaches explicitly model facial features to detect eye regions. In [11], dark regions are assumed to be associated with pupils and are searched within the face by iteratively thresholding the face image to locate eye candidates. Two eye candidates that satisfy certain anthropometric constraints are then identified as the detected eyes. This approach is reliable as long as subjects remain relatively frontal with respect to the camera and they don't wear eyeglasses. In [12]-[14], deformable eye templates are designed to fit the best representation of eyes in images. In this technique, eyes are detected through a recursive process that minimizes a specific energy function. In [12], the energy function is designed such that the energy is minimized when the total brightness inside the pupil candidate is small. While this method can detect eyes accurately, it requires the eye model to be initialized in close proximity to the eyes. Furthermore, it is computationally expensive and requires good image contrast. Since facial features exhibit large variability among subjects and under different experimental conditions, feature-based approaches tend to work well with some subjects in some experiments but can fail completely with other subjects under the same experimental conditions. Also, many feature-based techniques are limited to relatively frontal view of faces and therefore a more flexible and robust approach to eye detection is needed in order to cope with the expected variability of eye features for the full range of subjects' head movements.

Pattern-based approaches for eye detection are concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to detect the eyes. In [15]-[17], Eigen-Eye is developed to detect eye regions. Eigen-Eye uses Principal Component Analysis (PCA) to compute a set of basis images that provides a low dimensional representation of all possible eye images. To classify an image pattern as an eye or a non-eye, the image pattern is mapped to the space formed by the basis images and a similarity measure is used

to classify this pattern as an eye or a non-eye. The Eigen-Eye approach can fail to detect eye images that cannot be described by a simple linear combination of the basis images.

In [18] and [19], a large database of eye and non-eye images is used to train a Support Vector Machine (SVM) to detect eyes. One of the disadvantages of SVM is that the model designer has to make subjective decisions about the type of kernel to be used before the training procedure. This might not be ideal because it is difficult to justify quantitatively the choice of a specific kernel for a specific application.

In [20] and [21], Multi-Layer Feed forward Neural Networks are used to detect eye regions. Unlike Eigen-Eye and SVM, Neural Networks learn discriminative features about the eye from the training data. Since Neural Networks provide the best detection performance when the input data set exhibits large variability [22], a specific configuration of Neural Networks, Convolutional Neural Networks (CNN), was developed and optimized for the detection of eye regions in video images from the eye-tracker's camera.

II. Convolutional Neural Networks Overview

Convolutional Neural Networks was introduced by [23] and was specifically designed for adaptive image processing. It has been successfully applied in many practical applications [24]-[26]. CNN has three properties that are important for eye detection. Firstly, CNN is invariant to translation and robust to changes in scale and rotation [23]. Secondly, CNN emphasizes a key property of images which is that nearby pixels are much more likely to be correlated than more distant pixels. It achieves this property by extracting features that depend only on small sub-regions of the image. Information from such features is merged in later stages of processing in order to detect more complex features, and ultimately to yield information about the image as a whole. Lastly, in many applications that use Neural Networks [27]-[29], the original image is first preprocessed and the processed image is then fed into the Neural Networks for classification. This preprocessing step is essential, for example, for image intensity normalization. CNN does not require any preprocessing steps. It learns to build the preprocessing module and the classification module in a single integrated scheme.

A typical CNN is shown in Figure 2. It consists of a set of layers. Each layer contains one or more planes. A unit in each plane receives inputs from a small neighborhood in planes of the previous layer. Each plane can be considered as a feature map with a fixed feature detector that is convolved with a local window that is scanned over the planes of the previous layer. Multiple planes are usually used in each layer so that multiple features can be detected. Theses layers are called convolutional layers. Once a feature has been detected, its exact location is less important. Hence, the convolutional layers are typically followed by another layer, which does a local averaging and sub-sampling operation. These layers are called sub-sampling layers. Finally, the network has a fully connected feed forward Neural Network that carries out the classification task using the features extracted in the previous layers. The network is usually trained by a backpropogation gradient descent algorithm.

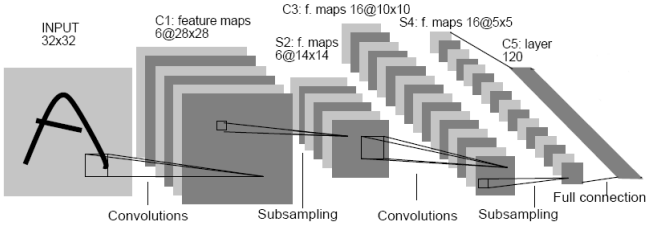


Figure 2: A Typical Convolutional Neural Network [23]

Three mechanisms that are unique to CNN are: (i) local receptive fields (sub-region connection), (ii) weight sharing, and (iii) sub-sampling.

In CNN, each unit in a feature map computes a weighted sum of inputs only from a small sub-region of the previous layer, and all the units in the feature map are constrained to share the same weight values as shown in Figure 3. Units in the feature map can be regarded as feature detectors and therefore all the units in a feature map detect the same pattern but at different locations in the input image. Due to the weight sharing, the operation of these units is equivalent to a convolution of a kernel comprising the weight parameters with the input image. This provides the basis for the invariance of the network outputs to translations of the input image. Since most often multiple features are needed for effective classification and detection, there are generally multiple feature maps in the convolutional layer, each having its own set of weights and bias parameters.

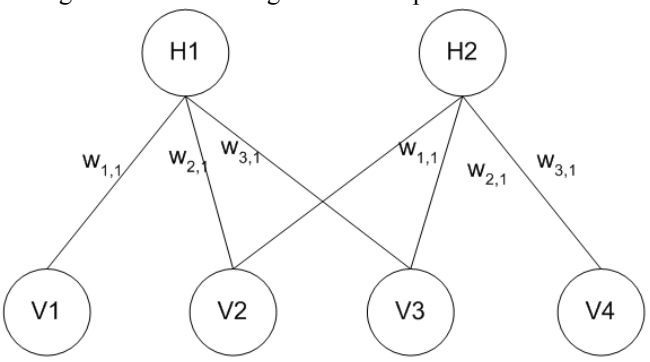


Figure 3: Each unit in layer H is connected by the same three weights to three units in the previous layer (V)

The outputs of the convolutional units are transformed by a sigmoidal non-linear activation function and are fed to a sub-sampling layer of the network. For each feature map in the convolutional layer, there is a plane of units in the sub-sampling layer. Each unit in the sub-sampling layer averages inputs from a region (for eye detection the region size is 2X2) in the corresponding feature map. This average is multiplied by an adaptive weight followed by the addition of a bias. The results are then transformed using a sigmoidal non-linear activation function. For eye detection the receptive fields in the sub-sampling layer were chosen to be contiguous and non-overlapping so that the number of rows and columns in the subsampling layer was half of the number of rows and columns in the convolutional layer.

III. CNN Architecture for Eye Detection

The CNN architecture for eye detection is shown in

Figure 4. To limit the complexity of the architecture, the CNN for eye detection uses only 2 stages of convolutional (C1, C2) and sub-sampling (S1, S2) layers. The first stage generally implements non-linear template-matching operation at a relatively fine spatial resolution, extracting basic features of the input image [23]. The second stage learns to recognize particular spatial combinations of previous features, generating complex features in a hierarchical manner.

In order to force different feature maps to extract different features in layer C2, each of the feature maps in layer C2 receives a different set of inputs from layer S1. It is expected that layer S2 will be able to extract a series of disjoint features of low-dimensionality that can be used for classification. Therefore, layer C3 is a fully connected layer where each unit is connected to all units of a single corresponding map in layer S2. Finally, all units in C3 are fully connected to form two outputs with softmax activation function

$$y_k = \frac{e^{a_k}}{\sum_{j=1}^{2} e^{a_j}} \quad (1)$$

where k=0 for eye images, k=1 for non-eye images and ak is the output unit activation k:

$$a_k = \sum_{j=1}^{M} w_{kj} z_j + w_{k0} \quad (2)$$

where w_{kj} and w_{k0} are network adjustable parameters for the output unit k and z_j is the output of the hidden unit j in layer C3, and M is the number of units in layer C3.

In the proposed architecture for eye detection, layer C1 is a convolutional layer with 4 feature maps. Each unit in each feature map is connected to a 5X5 neighborhood of the input image (size 36X36 pixels). The size of each feature map is 32X32 pixels, which is the result of the convolution of the 5X5 kernel (no zero-padding) with the input image. Each feature map has 26 adaptive parameters and the total number of parameters in layer C1 is 104.

Layer S1 is composed of four feature maps, one for each feature map in C1, and the size of each feature map in layer S1 is half the size of feature maps in layer C1 (16X16 pixels). Each feature map has 2 adaptive parameters and the total number of the parameters in layer S1 is 8.

Layer C2 is a convolutional layer with 15 feature maps. Each feature map receives inputs from a different subset of feature maps in layer S1. Each unit in each feature map is connected to a 3X3 neighborhood, at identical locations, in feature maps of layer S1. In total, the size of each feature map in layer C2 is 14X14 pixels and the total number of parameters in layer C2 is 303.

Layer S2 is composed of 15 feature maps. The size of each feature map is 7X7 and there are 30 adaptive parameters in this layer. Layer C3 has 15 units with each unit fully connected to all units of a single feature map in layer S2. Therefore, there are 750 adaptive parameters in this layer. Finally, all units in layer C3 are fully connected to form two outputs and there are 32 adaptive parameters in the final layer. In total, this architecture has 1227 parameters.

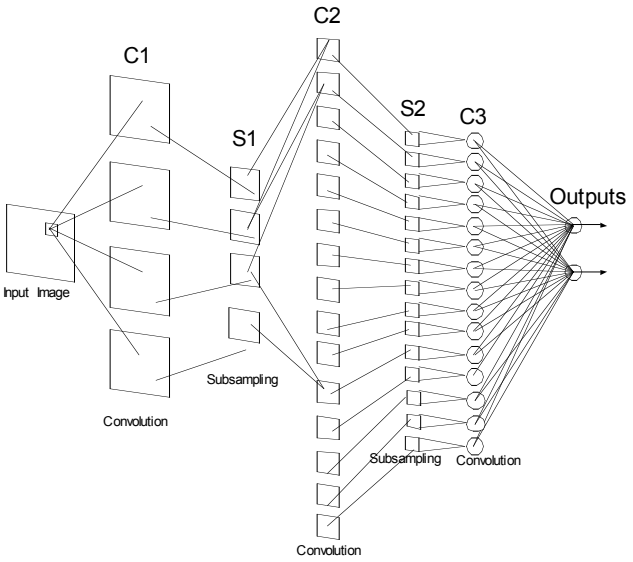


Figure 4: CNN Architecture for Eye Detection

IV. TRAINING METHODOLOGY

The training and validation dataset was built by manually cropping eye images from face images of 10 subjects. For each subject, 150 images of different head poses and face illuminations were collected. For each image, the portions of the image that contained the left and/or right eyes were cropped to fit an image size of 36X36 pixels. The total number of cropped eye images was 3000 images.

The generalization performance of the CNN depends strongly on the quality and the quantity of the training data [30]. To include large variety of experimental conditions (i.e. different face illuminations and different head poses), simulated images were created and were added to the original 3,000 images. To train the network with larger head rotations in the roll direction, rotated versions of the eye images were added to the training data. The degree of rotation was randomly selected between -30° to 30°. To train the network with larger variations in eye illumination, contrast and intensity transformations were applied to the original set of eye images. Using this strategy, the number of images in the dataset was expanded to 30,000 eye images. Some examples are shown in the first row of Figure 5.

To collect a representative set of non-eye images, a method described in [30] was adopted. Any portion of the cropped image obtained by the remote gaze estimation system that does not include an eye can be used as a non-eye image. The method aims to collect only non-eye images with high information value. The procedure stops when a total of 30,000 cropped (36X36 pixels) non-eye images were collected. Some examples of the collected non-eye images are shown in second row of Figure 5. The size of the dataset for eye and non-eye cropped images were 60,000.

Figure 5: Samples of eye and non-eye images

The training of CNN involves adjusting the network's

parameters, w, by fitting the network function y_k (equation 1) to the training data. This is done by minimizing an error function that measures the error between the network function, for any given value of w, and the training dataset. For the softmax activation function (Equation 1), the cross-entropy penalty function was used as the error function.

$$E(\mathbf{w}) = -\sum_{n=1}^{N} E_n(\mathbf{w})$$
$$E_n(\mathbf{w}) = -\sum t_{kn} \ln y_k(x_n, \mathbf{w}) \quad (3)$$

where n=1,…,N indicates the cropped images, $t_k \in \{0,1\}$ indicates the 2 classes of images (eye or non eye), and y_k are the network outputs.

The 1227 parameters of the CNN were trained by using the Stochastic Diagonal Levenberg Marquardt method (SDLM) [31]. SDLM attempts to estimate the learning rate for each network parameter so as to allow the network to converge faster.

To predict the generalization performance of the CNN, the "Early Stopping" procedure was used [32]. For the implementation of the "Early Stopping" procedure, the dataset was divided into a training dataset and a validation dataset. The training dataset included 50,000 images (25,000 eye and 25,000 non-eye) and the validation dataset included 10,000 images (5,000 eye and 5,000 non-eye). The performance of the CNN during the training session for both the training dataset and the validation dataset is illustrated in Figure 6. The training was stopped when the minimum error for the validation dataset was achieved (epoch 56, average cross entropy error of 0.024).

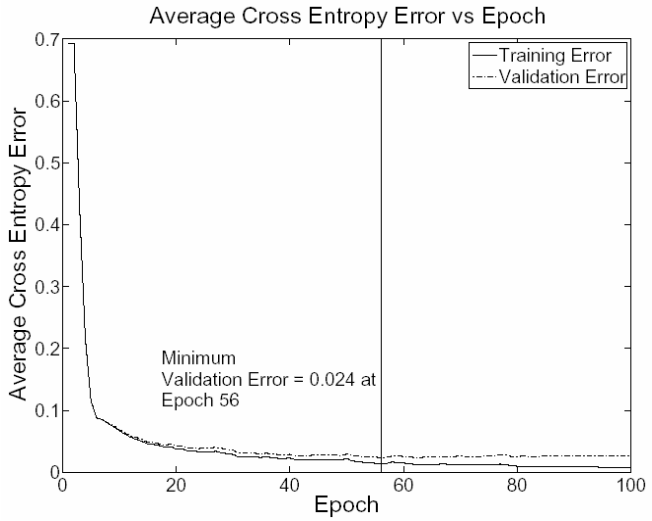


Figure 6: Training Session

V. System Performance

In the remote gaze estimation system, the regions of the image that were classified by the CNN as eye-regions are searched for specific eye features. If a non-eye region is classified as an eye-region these algorithms will fail to find the appropriate eye features and no point-of-gaze estimation would be generated for this region. Since false detection of eye regions by the CNN is rejected by the algorithms that search for specific eye features the point-of-gaze estimation system can tolerate high false alarm rate from the CNN. Based on experiments with 10 subjects the false alarm rate of the CNN can be as high as 10% without apparent degradation in the overall performance of the gaze estimation system. To determine the classification criteria for the CNN so that a false alarm rate of 10% is achieved, the Receiver Operating Characteristic curve (ROC) of the CNN classifier was constructed. Each point on the ROC was determined by applying a different threshold to the output of the CNN for the entire validation dataset. The result is plotted in Figure 7. As shown in Figure 7, a false alarm rate of 10% corresponds to a true positive rate (detection rate) of 99.3% (the actual network threshold is 0.45).

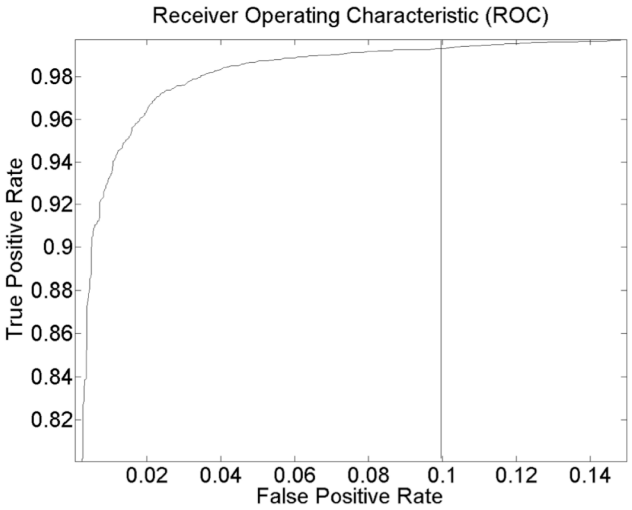


Figure 7: ROC Curve Used to Determine the Threshold

VI. Eye Localization Methodology

For each image from the remote gaze estimation system, the CNN generates a corresponding network response image. The value at each pixel in the network response image corresponds to the confidence of the network that the eye is present at this pixel location. The network response image is then compared with a specific threshold (0.45). Note that because the architecture of the CNN contains 2 sub-sampling layers (each sub-samples the image by a factor of 2), the number of pixels in the network response image is approximately 16 times smaller than the number of pixels in the input image. Pixels in the network response image that have values below the threshold are set to zero (Figure 8b). All network responses that are 4 pixels apart are clustered together to represent an eye candidate (Figure 8c) and the average CNN's network response for each cluster is computed. Finally, the center of each cluster is computed as the center of gravity of the position of pixels in the cluster weighted by the magnitude of their network response. The size of each eye window is similar to the size of the cropped images used for the training of the CNN (36X36 pixels). Since each image of the eye tracker's camera includes only one face (at most two eyes), only the eye windows that are associated with the two clusters with the highest network output are searched by the eye features algorithms (Figure 8d).

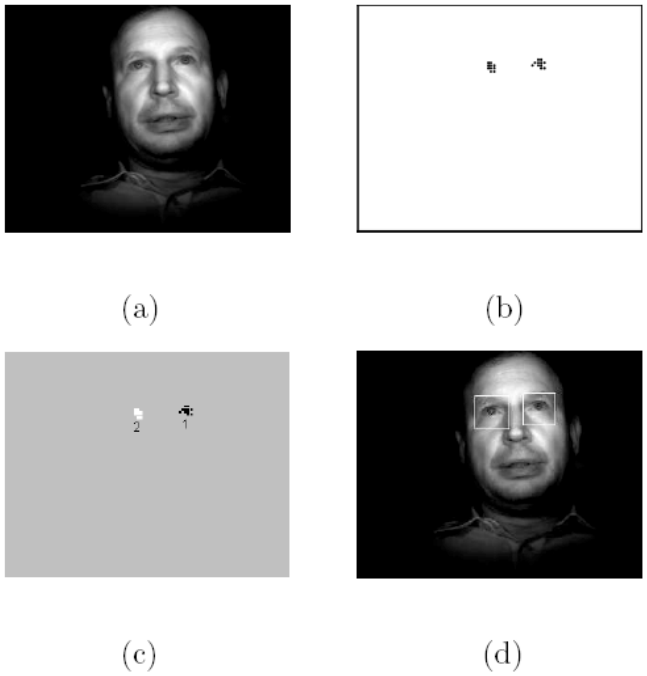


Figure 8: Eye Localization Algorithm (a) original image, (b) network response, (c) clustered network response, (d) eye windows

Based on the set of lenses and camera sensors that are used by the remote gaze estimation system, the pixel density can be changed by a factor of 2. This implies that in some optical configurations, the eye image can be as large as 72X72 pixels. To cope with different configurations without retraining the network for each specific optical configuration, images from the remote gaze estimation system are sub-sampled recursively 4 times (each time by a factor of 0.8) to generate 5 scaled images (including the original image) so that at least in one of the images, the eyes can fit within a window of 36X36 pixels (Figure 9). Each scaled image is then processed by the CNN and an image containing the network response is obtained. Pixels in each network response image with values lower than the specific threshold (0.45) are set to zero and the network response image of each scaled image is mapped back to the original image space. All network responses that belong to the same eye are clustered together. The center of the eye window for each eye candidate is then computed as the center of gravity of the position of pixels in the cluster, weighted by the magnitude of the network response at each pixel. The size of the eye window for each eye candidate is computed as the average of the eye-window-size that is associated with each pixel in the cluster and weighted by the magnitude of the network response at each pixel. Finally, the two clusters with the highest network output are considered by the eye-tracker's feature detector.

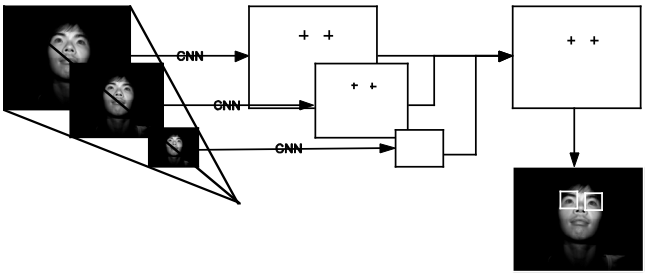


Figure 9: CNN Multiscale Analysis

VII. Experimental Results and Conclusions

The performance of the CNN for eye detection was tested in experiments with three subjects. Subjects were asked to move their heads within a volume of 20 x 20 x 20 cm^3 so as to simulate the expected range of head movements for a typical computer user. For each subject, 126 images at different head positions were collected. A total of 378 images were used for a test dataset.

To determine the performance of the CNN, the number of eyes that have been correctly detected and the number of false alarms were determined. An eye was correctly detected if and only if the detected window contained the full eye image. If the detected eye window only contained a partial or no region of the eye, it was counted as a false alarm.

For the test dataset the detection rate of the CNN was 100% for fully opened eyes. 4.8% of the eyes were either completely or partially covered by the subject's eyelids (i.e. due to eye blinks) and the CNN failed to detect those eyes. Since the training dataset had only eye images of fully opened eyes, the CNN had difficulty to detect eyes that were partially occluded. The false alarm rate of the CNN was $2.65X10^{-4}$ % for the test dataset. The low false alarm rate (even before the rejection of non-eye windows by the eye-feature algorithms) can be explained by the fact that in each image, the detector selected only the two eye-regions with the highest network response (i.e. maximum number of false alarms per image is 2). If one takes into account that: a) in each image from the remote gaze estimation system (1280X1024 pixels) there are approximately 81839 non-eye regions, and b) regions that include eyes have higher network output than non-eye regions, the number of false alarms per image is much smaller then the expected false alarm rate when all the sub-regions with outputs greater than the threshold (0.45) are selected.

Figure 10 and 11 show that the CNN is robust for large head movements (changes in scale and orientation) and changes in face illumination. The experimental data suggested that the CNN could detect eyes for head movements that span the expected range of head movements when subjects use a computer system.

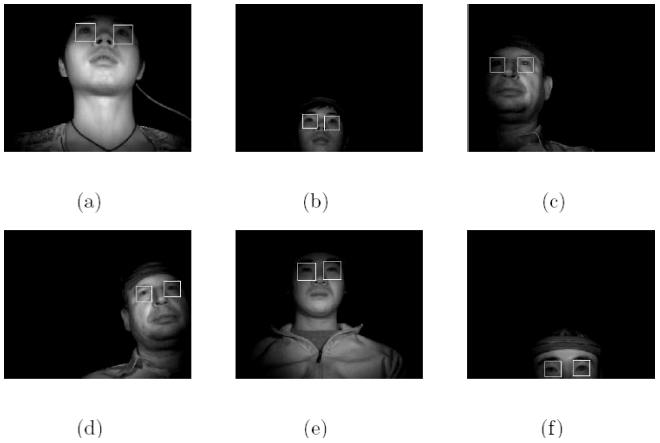


Figure 10: Experimental Results A

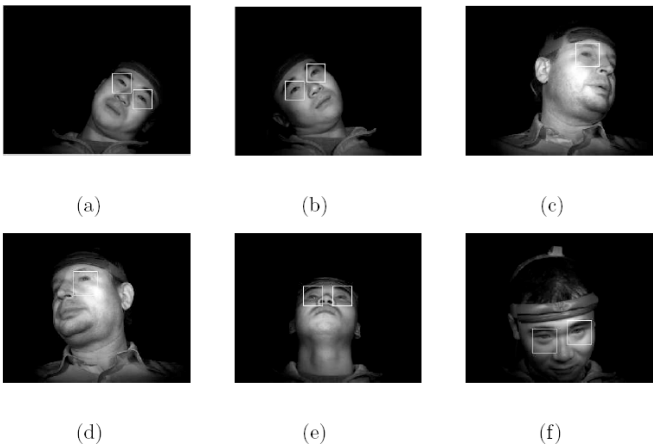


Figure 11: Experimental Results B

REFERENCES

[1] M. Eizenman, L. H. Yu, L. Grupp, E. Eizenman, M. Ellenbogen, M. Gemar, and R. D. Levitan, "A naturalistic visual scanning approach to assess selective attention in major depressive disorder*," Psychiat. Res*., vol. 118, no. 2, pp. 117-128, May 2003.

[2] M. Eizenman, T. Jares, and A. Smiley, "A new methodology for the analysis of eye movements and visual scanning in drivers," in *Proc. 31st An. Conf. Erg. & Safety*, Hall, Quebec, Canada, 1999.

[3] J. L. Harbluk, I. Y. Noy, and M. Eizenman, "The impact of cognitive distraction on driver visual and vehicle control," in *Proc. Transp. Res. Board 81st An. Meet.*, Washington, DC, USA, Jan. 2002.

[4] D. Cleveland, "Unobtrusive eyelid closure and visual point of regard measurement system," in *Proc. Tech. Conf. on Ocular Measures of Driver Alertness*, sponsored by The Federal Highway Administration – Office of Motor Carrier and Highway Safety and The National Highway Traffic Safety Administration – Office of Vehicle Safety Research, Herndon, VA, USA, 1999, pp. 57-74.

[5] P. A. Wetzel, G. Krueger-Anderson, C. Poprik, and P. Bascom, "An eye tracking system for analysis of pilots' scan paths," *United States Air Force Armstrong Laboratory*, Tech. Rep. AL/HR-TR-1996-0145, Apr. 1997.

[6] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *Int. J. Ind. Erg.*, vol. 24, no. 6, pp. 631-645, Oct. 1999. Paraná, Entre Ríos, Argentina, in 2000.

[7] R. Sharma, V. I. Pavlovi , and T. S. Huang, "Toward multimodal human-computer interface," *Proc. IEEE*, vol. 86, no. 5, pp. 853-869, May 1998.

[8] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey, "Human-computer interaction using eye-gaze input*," IEEE Trans. Syst., Man, Cybern.,* vol. 19, no. 6, pp. 1527-1534, Nov./Dec. 1989.

[9] L. A. Frey, K. P. White, and T. E. Hutchinson, "Eye-gaze word processing," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 4, pp. 944-950, Jul./Aug. 1990.

[10] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 1124-1134, 2006.

[11] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking eyes and monitoring eye gaze," in *Proceedings of the Workshop on Perceptual User Interfaces*, 1997, pp. 98-100.

[12] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal Computer Vision*, vol. 8, pp. 99–111, 1992.

[13] X. Xie, R. Sudhakar, and H. Zhuang, "On improving eye feature extraction using deformable templates," *Pattern Recognition*, vol. 27, pp. 791–799, 1994.

[14] L. Kin-Man and Y. Hong, "Locating and extracting the eye in human face images," *Pattern Recognition*, vol. 29, no. 5, pp. 771–779, May 1996.

[15] Q. Ji and X. Yang, "Real time 3D face pose discrimination based on active IR illumi- nation," in *Proceedings of the 16th International Conference on Pattern Recognition* (ICPR'02), vol. 4, 2002, pp. 310–313.

[16] W. Huang and R. Mariani, "Face detection and precise eyes location," in *Proceedings of the International Conference on Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 722–727.

[17] W. Huang, Q. Sun, C. P. Lam, and J. K.Wu, "A robust approach to face and eyes detection from images with cluttered background," in *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1. Washington, DC, USA: IEEE Computer Society, 1998, pp. 110–113.

[18] A. Cozzi, M. Flickner, J. Mao, and S. Vaithyanathan, "A comparison of classifiers for real-time eye detection," in *Proceedings of the International Conference on Artificial Neural Networks*, 2001, pp. 993–999.

[19] J. Huang, X. Shao, and H. Wechsler, "Face pose discrimination using support vector machines," in *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1. IEEE Computer Society, 1998, p. 154.

[20] M. Motwani, R. Motwani, and F. C. Harris, "Eye detection using wavelets and ANN," in *Proceedings of Global Signal Processing Conferences and Expos for Industry*, 2004, pp. 27–30.

[21] H. Peng, C. Zhang, and Z. Bian, "Human eyes detection using hybrid neural method," in *Proceedings of 4th International Conference on Signal Processing*, 1998, pp. 1088–1091.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computing and Applications*, vol. 18, pp. 1527–1554, 20ß06.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, 1998, pp. 2278–2324.

[24] B. Fasel, "Robust face analysis using convolutional neural networks," in *Proceedings of the International Conference on Pattern Recognition (ICPR 02)*, vol. 2, 2002, pp. 40–43.

[25] D. D. Lee and H. S. Seung, "A neural network based head tracking system," in *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems*, vol. 10. Cambridge, MA, USA: MIT Press, 1998, pp. 908–914.

[26] [57] S. J. Nowlan and J. C. Platt, "A convolutional neural network hand tracker," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. The MIT Press, 1995, pp. 901–908.

[27] R. Feraud, O. J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks*," IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, 2001.

[28] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[29] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. IEEE Computer Society, 1997, pp. 130–136.

[30] D. Beymer, A. Shashua, and T. Poggio, "Example based image analysis and synthesis," *Massachusetts Institute of Technology*, Tech. Rep., 1993.

[31] S. Becker and Y. LeCun, "Improving the convergence of back-propagation learning with second-order methods," in *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds., 1989, pp. 29–37.

[32] W. S. Sarle, "Stopped training and other remedies for overfitting," in *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, 1995, pp. 352–360.