

Monitor Eye-Care System using Blink Detection A Convolutional Neural Net Approach

Gautam Worah
Student at Computer Dept.,
S.P.I.T

Mehul Kothari
Student at Computer Dept.,
S.P.I.T

Aasim Khan
Student at Computer Dept.,
S.P.I.T

Meghana Naik
Professor at Applied Sciences and Humanities Dept.,
S.P.I.T

Abstract—We spend most of our time in front of digital devices, be it at home, at work or while travelling. When working with screens is important to blink regularly, failure of which can result in a condition known as Computer Vision Syndrome(CVS). To avoid the above situation, we can create a system which involves blink detection and if the rate falls below the normal threshold the person is warned immediately. This paper overviews different algorithms used in blink detection. The various phases in this system are image capture, face detection, eye localization and finally blink detection. The standard algorithms for each phase have been discussed. We have proposed a method for blink detection using convolutional neural network to detect the eye states and predict the blinks. Finally, a comparison of our method is made to the other methods of detecting blinks in both still images and live video feed scenarios. Regular usage of the proposed system could significantly reduce the signs of computer vision syndrome and ensure efficient work experience for people with long term computer use.

Keywords— Eye-Care; Face Detection; Blink Detection; Convolutional Neural Network;

I. INTRODUCTION

Computer Vision Syndrome(CVS) is a condition which is caused by focusing on a computer or any other display device for long intervals of time. Some of the symptoms include a headache, blurred vision, irritated eyes, double vision and much more. It has been demonstrated that blinking reduces up to 60% while using digital devices which is a major cause for CVS.

Poor lighting conditions can make matters worse. One of the major symptoms of CVS is dry eye which is a condition when eyes dry up accompanied by redness, burning, eye irritation and eye fatigue. Dry eye is affected by many internal and external factors including Reduced blink rate, incomplete blinking, usage of contact lens and dysfunction of glands.

Dry eye could be avoided by taking up measures like having a minimum distance of 20cm between the user's eye and screen, adjusting the viewing angle to 15 degrees lower than horizontal level and a minimal refresh rate of 75hz. Room lighting and screen lighting should be checked before using the computer. One of the most important measures is to blink at a normal rate and avoid drying up of tears.

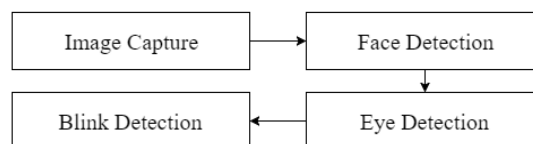
This paper aims to provide the reader with a comprehensive review of the blink detection algorithms. The paper compares the different techniques and systems. The authors hope that this

paper will act as a guide for researchers and users of these systems.

The methods involved could be either through expensive hardware or simple hardware having the optimized software. The former method includes infrared cameras and illuminators but the latter involves a simple camera with algorithms going through the phases of face detection, eye detection and then finally blink detection. All the algorithms discussed primarily use the more passive method.

II. PROPOSED SYSTEM

Eyeblink detection is made up of four phases. (1) Image Capture (2) Face detection (3) Eye localization and finally (4) Blink Detection.



All stages are discussed individually and possible algorithms and techniques used in particular stages have been discussed.

A. Image Capture

Image capturing is done in real time online mode. Webcams mounted on the monitor screens, inbuilt laptop webcams and front camera of mobile phones will be used for the purpose. A real blink of an eye takes 300 to 400 milliseconds. Since there're 1000 milliseconds in each second, a blink of an eye takes around one-third of a second. Though it seems like a short period, considering the span of a single second it is significant enough. The standard webcams and front cameras usually have recording rates of 30fps with some going up to 120fps to 240fps. For this system, a capturing device of 30fps is well suited.

B. Face Detection

Face detection is the first stage in blink detection technique. While getting to the core of the matter i.e. to detect the blink rate we first have to identify the peripheral structure which is the face in this situation. It is a methodology which involves detection of a human face from visual media. Researchers have tried to come up with a number of algorithms and techniques which serve the above purpose. Few of the important ones are mentioned below.

1) *Viola Jones face detection algorithm*: This algorithm was introduced by Paul Viola and Michael Jones in 2001 which was the first algorithm to detect objects in the real-time

environment. It only manages to detect objects but not recognize them and was primarily introduced to detect faces[14][15].

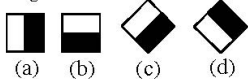
a) *Haar-like feature*: A window is placed on the object and Haar-like feature calculated for various parts of the face.

b) *Integral Image*: Illustration of the original image in this form which allows computation of features in an efficient and faster manner.

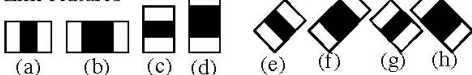
c) *Adaboost Training*: It is used to select the features and train the classifier.

d) *Cascaded classifier*: Rejection of negative subwindows and detection of positive instances.

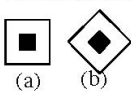
1. Edge features



2. Line features



3. Center-surround features



4. Special diagonal line feature used in [3,4,5]



Fig. 1. Window placed on various features

Advantages of this method include high accuracy and high computation time. But one of the disadvantages is that it cannot detect black faces[13].

2) *Local Binary Pattern (LBP)*: This is a relatively new approach introduced by Ojala. This technique involves dividing the image into small pieces and then extracting features from it. The features obtained from a histogram which gives the representation of a single image. These histograms can then be used to compare different images. An important aspect of LBP is uniformity. A Local Binary Pattern is uniform if it shows two bitwise transitions from 0 to 1 or vice versa. In the method of LBP, we consider one point and eight other points around it ie in a circular fashion(as shown in the figure). 00000000 and 11111111 are examples of patterns with zero transitions. 11000011 is an example of a pattern with 2 transitions. There are two benefits of using uniform LBP. The first one being the usage of memory is lesser and the second one is its ability to spot only important features.

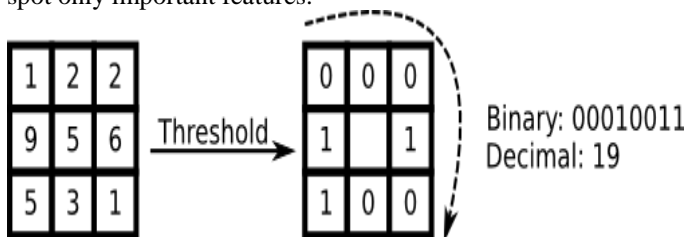


Fig. 2. Local Binary Pattern using Threshold

One of the advantages includes its simplicity and its ability to describe image text feature. Disadvantages mainly include its focus only towards binary and grey images.

3) *Histogram of Oriented Gradients(HOG)*: HOG is a feature descriptor used for the purpose of object detection. The occurrences of gradient orientation are counted in the input image which has been divided into 16x16 cells. Usage of HOG became famous by Navneet Dalal and Bill Triggs. Steps followed in this technique are:

- a) Make the image black and white
- b) Looking at every pixel compare it with the surrounding pixels and draw an arrow in the direction of increased darkness.
- c) Follow this step for all the pixels and these arrows are termed gradients.
- d) Divide the image into 16x16 dimensions cells and in each cell, we analyze the direction of gradients.
- e) All the gradients in the cell are replaced with the maximum number of gradients in a particular direction.
- f) We finally get a simple HOG image. We compare parts of this image to the already trained HOG pattern which represents a face.

C. *Eye Localization*

For eye localization we'll be using a facial landmark extractor implemented inside DLib produces 68 (x, y)—coordinates that map to specific facial structures. The library implements the paper from Vahid Kazemi et al. [4]. These 68 point mappings were obtained by training a shape predictor on the labeled iBUG 300-W dataset. After getting these 68 points, the points from 37-42 represent the right eye and the points 43-48 represents the right eye. Using the 2 sets of points we create a rectangular cropped image of the 2 eyes. This entire process is called eye localization.

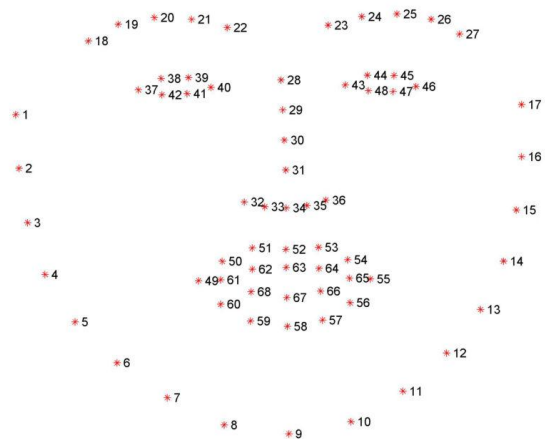


Fig. 3. Extraction Points from DLib Library

The third step involves binarization of the localized image. This step is not needed for all the algorithms we have tested for blink detection however, some need them. The RGB image is converted to a binary image of 0's and 1's. It is done mainly to reduce the time consumption and simplify the execution of some algorithms.

The final step involves determining the status of the eye. The 2 states open and closed will help us determine whether there was a blink or not. Some algorithms will involve comparing the current frame with the previous frames or a standard frame to detect the state while others will analyze each frame

individually. In the latter, however, to identify the blink we have to compare the current frames with previous ones.

C. Blink Detection

1) *Eye Aspect Ratio*: This algorithm takes the facial points extracted using the Dlib library[4] to detect the eye blinks. Each eye is represented using 6 points and there's a correlation among these points. Based on the work done by Soukupova et al. we can derive an equation known as eye aspect ratio (EAR).

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

The EAR equation computes the distance between the vertical eye landmark in the numerator and denominator contains the distance of the horizontal landmark multiplied by 2 to weigh in the appropriately. EAR ratio is approximately constant when the eye is open and it falls to zero when the eye is closed. Using this we can detect when a blink has taken place. The algorithm computes EAR for every frame from the capturing device and then flags every change in state. 2 successive changes are taken into account (open to close and close to open) and are counted as a single blink.

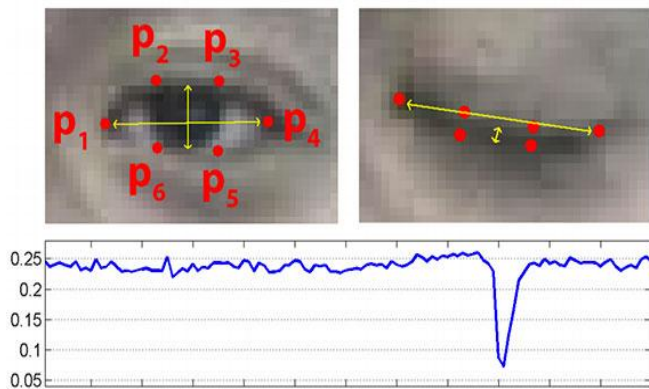


Fig. 4. EAR in Demonstration

2) *Template Matching*: Here the cropped localized eye images are taken for computation. These eye cropped images are referred as eye regions. The algorithm takes in account of the 2 most recent frames the previous frame and the current frame. These 2 images are tracked to determine whether the eyes are open or not. The previous frame is subtracted from the current frame to get the initial frame difference. Then a binary thresholding is used to clearly show the region of change around the person's eye. Black colour indicates a negligible change in the pixel. Significant changes will be denoted by a white pixel. If there isn't any significant difference then the eyes are considered to be open. We'll now save this template as an open eye template. The eyes regions are 80% width and 60% height of this template. This standard is maintained as the template will be compared with all the future eye frames. The templates are refreshed after every 150 frames or 5 seconds to adjust to the changing environment.

$$R(x, y) = \frac{\sum_{x',y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}}$$

Fig. 3. Normalized squared difference for template matching.

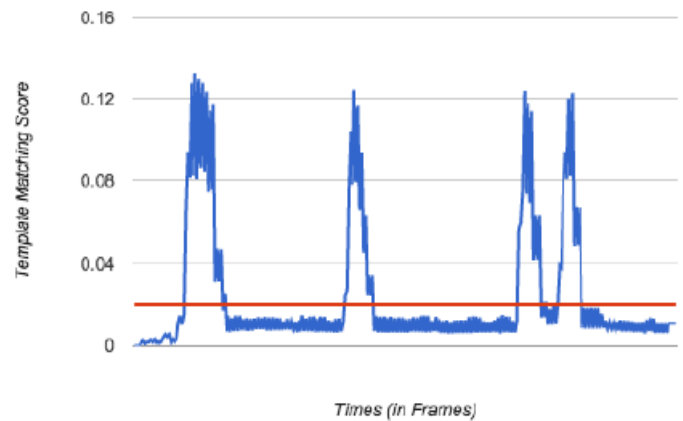


Fig. 5. Proper Template Matching

To match the real-time frame with the saved open eye template we use normalized squared difference [16]. This operation yields a matrix of values from 0 to 1. A 0 indicated that the 2 pixels were almost identical whereas 1 indicates the greatest point of difference between the template and the image. The smallest value from this matrix will indicate the similarity score for template matching. Again a threshold is fixed and if it is crossed we can say that eye has changed state.

III. IMPLEMENTATION USING CONVOLUTIONAL NEURAL NETWORKS

A convolution Neural Network is trained on the RPI ISL Eye Database. From the entire dataset of 3843 images, approximately 80% of the dataset is used for training the CNN and remaining 20% data will be used for validation. This training set is augmented using common image transformations where it is randomly sheared, shifted, flipped and zoomed. The augmentation factor was 24 times, which formed the final training set of approximately 73000 images.



Fig. 6. Training Set Augmentation

To prevent overfitting, we'll use a small CNN which will have a limited entropic capacity. Such a convnet will have fewer chances of storing irrelevant features and future prevent overfitting. Our convnet have few layers and filters with dropout. Dropout also helps with overfitting, by preventing a layer by seeing the same pattern twice. To greatly improve the accuracy, we'll also use a pre-trained neural network called VGG-19. VGG-19 is trained on the ImageNet dataset and contains features that'll help us get a better accuracy [14]. We have computed the model on our training and validation set once and then train our original convnet on top of it. Fine Tuning with a very small learning rate is performed on the top layer provided by us. This model is loaded and used in our application where we'll be feeding it with localized eye images from the webcam. Like the other algorithms, here our application will predict whether the eye is open or closed and flag every transition from open to closed state. One successful cycle will count as blink.

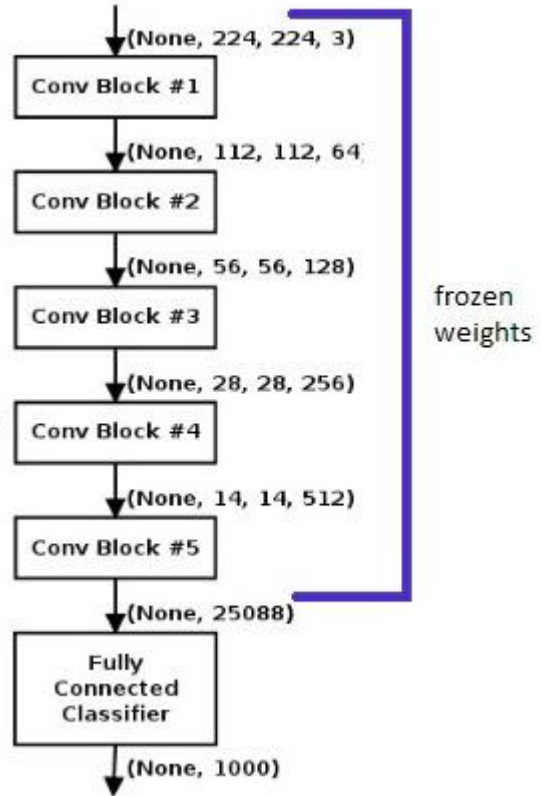


Fig. 7. Structure of our CNN where the last layer represents our classifier

IV. PERFORMANCE EVALUATION

A. Still Image Performance

The performance of the above algorithms is evaluated on the still images of RPI ISL Eye Database [2]. This database includes the cropped image of left/right eye (open/closed) region of different sizes and orientations. The template matching algorithm won't work in this case as it directly extracts templates from a live feed and won't work with still images. There is a total of 3843 eye images with 2070 open eyes and 1773 closed eyes.

Algorithm	Training Set		Validation Set			
	Closed Eyes	Open Eyes	(Closed Eyes)	(Open Eyes)		
Metrics			Samples	Accuracy	Samples	Accuracy
Eye Aspect Ratio	-	-	2070	92.53%	1773	91.67%
Template matching	-	-	-	-	-	-
CNN	1656	1418	414	99.2%	355	98.76%

Results based on RPI ISL Eye Database

Table 1: Still Image Performance

Experiment Scenario	Actual Blinks	Eye Aspect Ratio		Template Matching		Convolution Neural Network	
		Detected	Accuracy	Detected	Accuracy	Detected	Accuracy
Frontal view with glasses	131	138	94.65%	126	96.18%	130	99.23%
Frontal view without glasses	60	64	93.33%	53	88.33%	60	100%
Upward view	70	72	97.22%	66	93.93%	71	98.57%
All scenarios	261	267	98.08%	245	93.86%	263	99.23%

Results based on ZJU Eyeblink Database

Table 2: Live Feed Performance

V. CONCLUSION

This system should play an important role in preventing Computer Vision Syndrome. Out of all the blink detection algorithms, we have tried, our proposed CNN algorithm works the best. This performance is directly visible both in still image and video feed dataset with our algorithm achieving near perfect results. This deep learning CNN trains fast in a GPU based system and can be deployed quickly. The main objective of the blink detection is to only find whether the blink rate is low and not the exact blink rate. Therefore, improvement in accuracy in the future iteration of this systems won't directly affect the performance of the system. However, such improvements can be useful in the implementation of applications mentioned in the future scope section of this paper.

VI. FUTURE WORK

In addition to the low blink rate and fatigue detection, the system can be also extended to incorporate user authentication, where the system detects a liveliness of the person. Blink detection is known to have a variety of applications, including but not limited to fatigue detection, driver alertness measure etc. This type of protection is used in systems where face-detection is used to authenticate users. Regarding the blink prediction part, we can train an SVM based classifier which helps us reduce the false positives. This classifier will be trained on a window of n frames with the centre frame being the current frame. Such a classifier will help us determine the false positive of the blink detection algorithms due to a rapid head movement where the eye might appear closed. However, for the current system, such a high accuracy isn't needed. This addition will be useful when the current system will be used for extended applications.

REFERENCES

- [1] Pan, Gang, Lin Sun, Zhaohui Wu, and Shihong Lao. "Eyeblick-based anti-spoofing in face recognition from a generic webcam." In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1-8. IEEE, 2007.
- [2] Wang, Peng, and Qiang Ji. "Learning discriminant features for multi-view face and eye detection." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 373-379. IEEE, 2005.
- [3] Wang, Liting, Xiaoqing Ding, Chi Fang, Changsong Liu, and Kongqiao Wang. "Eye blink detection based on eye contour extraction." In *Image Processing: Algorithms and Systems*, p. 72450. 2009.
- [4] Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867-1874. 2014.
- [5] Chau, Michael, and Margrit Betke. *Real time eye tracking and blink detection with usb cameras*. Boston University Computer Science Department, 2005.
- [6] Morris, Tim, Paul Blenkhorn, and Farhan Zaidi. "Blink detection for real-time eye tracking." *Journal of Network and Computer Applications* 25, no. 2 (2002): 129-143.
- [7] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [8] Raina, Rajat, Andrew Y. Ng, and Daphne Koller. "Constructing informative priors using transfer learning." In *Proceedings of the 23rd international conference on Machine learning*, pp. 713-720. ACM, 2006.
- [9] Lienhart, Rainer, Alexander Kuranov, and Vadim Pisarevsky. "Empirical analysis of detection cascades of boosted classifiers for rapid object detection." *Pattern Recognition*(2003): 297-304
- [10] Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. "Face recognition with local binary patterns." *Computer vision-eccv 2004* (2004): 469-481
- [11] Tefft, Brian C. Prevalence of motor vehicle crashes involving drowsy drivers, United States, 2009-2013. Washington, DC: AAA Foundation for Traffic Safety, 2014.
- [12] Bhide, Nimish, et al. "VEHICLE COLLISION PREVENTION SYSTEM."
- [13] Chauhan, Mayank, and Mukesh Sakle. "Study & Analysis of Different Face Detection Techniques." *International Journal of Computer Science and Information Technologies* 5.2 (2014): 1615-1618.
- [14] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2001.
- [15] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." *International journal of computer vision* 57.2 (2004): 137-154