A Low Cost Eye-tracking Interface to Help the Physically Impaired Users

Thomas Karpat (SCPD)
Stanford University
tkarpati@stanford.edu

Shankar Thadhalani (SCPD)
Stanford University
shankart@stanford.edu

Abstract

This paper presents a low cost alternative to traditional gaze-tracking systems to help pysically impaired uses interact with machines. We propose a passive system which can utilize the integrated cameras already present in most laptops and tablets available today. The system utilizes face and eye tracking to locate the user in the camera frame and a convolutional neural network to infer the point-of-gaze of the user from the captured image from the integrated camera. Due to the passive nature of our system, it doesn't require the expensive active external lighting sources required by most existing solutions. The system also doesn't require the user to remain in a fixed position or to immobilize their head to use the system.

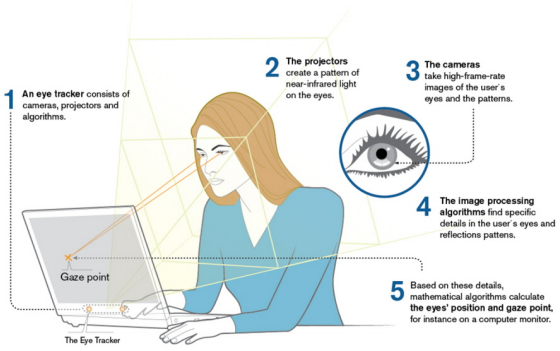

Figure 1. Existing solutions to gaze-tracking problem

1. Introduction

A gaze-tracking system has numerous application in human computer interfaces, from use as an input modality for the imparired[1], to determination of region of interest for augmented reality and image processing. The goal of gaze-tracking is to determine the location that a user is looking at. For aiding the impaired, the system would determine the location on the screen that a user is looking, and then move the cursor to that location. This can be used as a form of input for those that cannot move a mouse or use a keyboard. This can then be used with in on-screen keyboard to interact with a system.

The system proposed is a completely passive system that uses a single camera. The system attempts to eliminate the need for expensive external equipment, just using the integrated camera available on most laptops and tablets. The system is composed of two parts. The first part is to determine the location of the user in the frame of the camera, and the location of the eyes. This information is then fed into a convolutional neural network to determine the point of gaze of the user from the information determined from the camera image.

Beyond the location of the eye within the eye socket, the system must also determine the head pose since the user is not required to have their head constrained or immobilised, and as such the location and pose of the head is not known a priori. This requires that the convolutional network must also be passed the location of the eyes in the frame and the location of the eyes relative to the face to effectively be able to determine the pose of the users head.

2. Existing Solutions

Many existing systems rely on external devices[11][12], or require the user to have their head immobilized or wear a camera attached to their head. This makes eye tracking an expensive and unreachable solution for many. Some of these are based on infrared light sources and rely on the reflection of the light from the user's pupils to identify the point of user interest. Our system hopes to work around the use of these external devices. A brief introduction of existing solution mechanism can be seen Figure 1.

The literature contains systems that are built with strictly classical approaches to those that are built on more modern neural networks for processing[3][4][5][6][7][8]. Our system hopes to use both methods where we can preprocess the data that is fed into the network using classical methods to aid in the processing. We hope to leverage the existing dataset that was produced by the group at University

of Texas, Arlington[4]. Their work contains several videos with synchronized homographies that link the location and pose of the eyes with the location of the screen relative to each other. This solution however requires a user to have a head mounted camera to capture a video of the eyes moving in the users eye sockets. While this may be effective, it can be cumbersome for the user.

3. Technical Approach

TK – ?? Is this still correct? The system is designed to run 2 threads, one thread to capture and validate input from webcam and another to process the captured input. The second thread processes the captured and sanitized input through our convolutional neural network and obtains the point of user interest on the screen. In following section we explain the break down of system from image capturing to evaluating point of interest.

3.1. Image capture and face/eye detection

TK – Need more information on face and eye detection. diagrams, equations, etc...

We implemented the system in Python 2.7 using the OpenCV libraries for image processing tasks. The use of OpenCV gives us an easy hardware abstraction layer to video capture and display peripherals. We use haar-cascade utilizing Viola-Jones face recognition algorithm to identify face and eyes.

First we capture input stream using peripheral interface provided by OpenCV. We convert each frame as it is captured to grayscale, this frame is then analyzed using haar-feature based cascade classifier to identify a face. Once we have detected a face, we process frame detected as face again using haar-feature based classifier to identify eyes.

In our initial runs we encountered difficulties with these cascade classifiers. This can be seen in (Figure 2). In order to sanitize the inputs passed to our neural network, we added additional defensive coding to eliminate false positive detection for eyes or faces. We make sure a detected face has eyes and location of the eyes are not in the lower half of the detected face

Once the face itself has been extracted from the video and the eyes are isolated from the face, then images of the face and the location of eyes are fed into a neural network to determine the location of the gaze.

3.2. Eye Image Preprocessing

To aid the neural network, the input data images are preprocessed. We perform the preprocessing step to aid in locating the pupils withing the images of the eyes. The pupils of the eyes are distinctive in that they are, in general, the only large round thing in the image. Also, due to the fact that the iris tends to be of a darker color than the sclera and

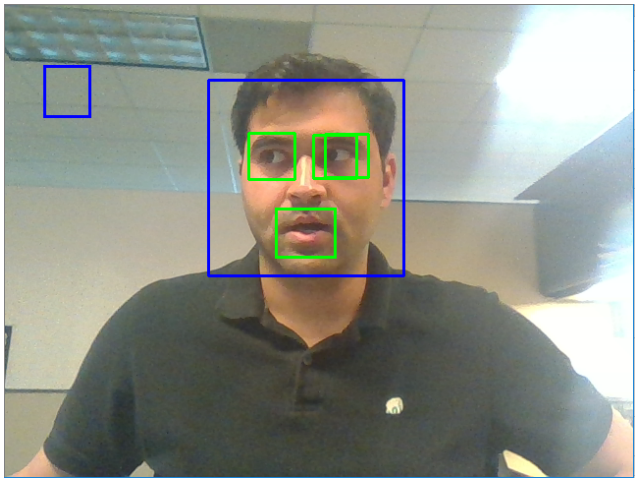

Figure 2. Failure observed without defensive coding for face and eye verification.

should be easily detectable in a grayscale image of the eye when looking at the gradient of the image. The pupils are attempted to be located by performing a search for circles in the Hough space that are of a certain size. The pupils are then added as a mask to the eye image as a fourth channel in addition to the RGB images of the eyes. This information is provided as a hint to the neural network for processing the image. The search for pupils is limited to circles to filter out false matches. The radius of the pupil in the image is limited to a range between 1/10 to 1/4 relative to the bounding box size. The range is computed relative to the bounding box sizes to account for changes in size of the detected eyes in the captured frames that may account from the user being closer or farther from the camera.

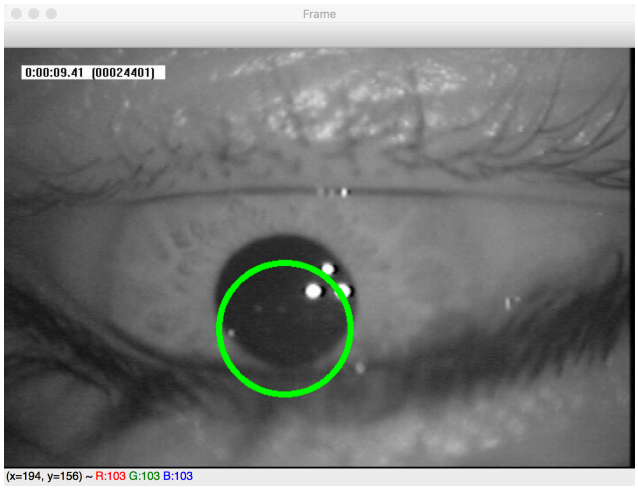

Figure 3. Detection of pupils using Hough space search

We don't only pass the pupil mask to the neural network

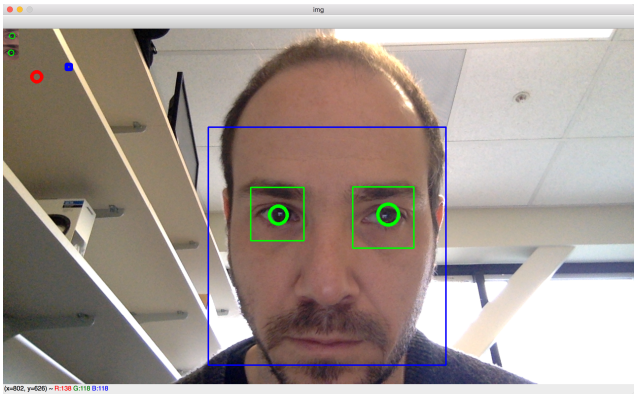

Figure 4. Detection of pupils during online training and inferrence

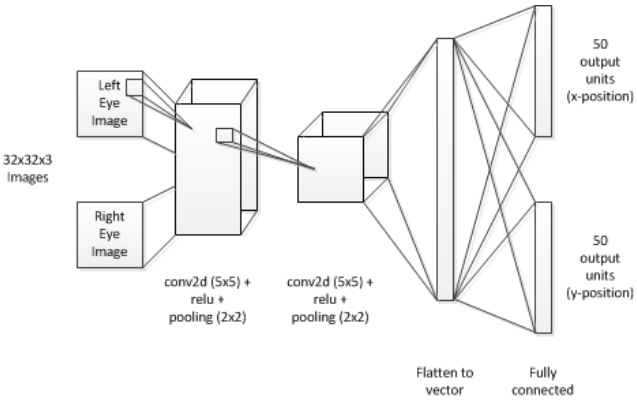

Figure 5. Neural Network Architecture

because the search in Hough space can result in either more or less circles in the eye image, or may be offset slightly from the actual pupils. To counteract this, we provide both the RGB image of the eye and the mask that we determine locating the pupils. The detection of the pupil for processing the data set is shown in Figure 3. During online training and testing we also perform a hough space search for pupils in each image of the eye. The results for this can be seen in Figure 4 which show the capture and processing the pupils

3.3. Convolutional Neural Network

Our neural network is implemented in the TensorFlow framework. The network is a multilayer classifier where the inputs are the extracted images of both the left and the right eyes and the bounding boxes for both eyes and the face in the original image. The two images of the eyes are 32x32x4 images where the extracted images of the eyes are scaled to a set size. This gives us a fixed image size to feed into the network invariant of the original size of the eyes in the original video frame. The size of 32x32 should be large enough to provide enough detail without creating a network that is too large. It is in line with previous work.

The images are fed into a bank of 16xs convolutional filters. This results in an output of 32x32x16 outputs. The filter kernels have support of 5 pixels in the x- and y-axes. The output of the convolutional filters is passed through a rectified linear activation function. Following the convolution layer, we perform max pooling with a non-overlapping 2x2 filter to reduce the dimentionality.

A second bank of convolutional filters is then convolved with the output of the first bank. The second set of layers is also a convolution and max pooling layer. The input of the second bank has shape of 16x16x16 inputs. The second bank has 16 input and 32 output channels and stride of 1. The filter kernel has support of 5 pixels. This is essentially the same as the first layer with a rectified linear activation function, but the dimentionality is smaller in the image x- and y-axes, and larger in the filter bank depth. The output again is passed through a max-poolong layer.

Following the two convolutional layers, the output is flattened into a single vector and passes through a fully connected layer of 200 units. This layer is then connected to two 50 unit output layers in fully connected fashion. These two sets of 50 units represent the x- and y- outputs of the gaze. The two sets of 50 output represent segmenting the angle of gaze into 50 segments where the center output unit represents a gaze of 0 degrees. The x- and y- axes are separate here and the output is a softmax function and represents the probability distribution of the output in the x- and y- directions.

4. Material to be integrated

We propose to implement an eye-tracking system utilizing a single camera. Gaze tracking has numerous application in human computer interfaces from use as input for the impaired[1], to determination of region of interest for augmented reality and image processing. Our system would use a camera on a laptop and determine the target of the users gaze. This would then be used to move a cursor around a graphical keyboard on the display which could be used for input. Our design results in a multi-segment system. We propose to first segment the face from the image. Then the left and right eyes are each segmented from the face. The eyes are then fed into a trained neural network to determine the location of focus for the user. We will examine work that has already been done in the past on gaze tracking to build upon. The literature contains systems from strictly classical approaches to systems that are built on neural networks for the processing We hope to be able to use a combination of the two to achieve good results. While existing commercial solutions use external light source and additional hardware to implement eye tracking, we expect to utilize openly available gaze eye-tracking dataset to eliminate the requirement for any additional hardware.

5. Problem Statement

The existing commercial solution available require external hardware being attached to existing device[11][12] This makes eye tracking an expensive and unreachable solution for many. These solutions are based on external infrared light source and rely on the reflection of the light from the users pupil to identify point of user interest. We hope to utilize the built in web camera of our system and eliminate the necessity of additional hardware such as external light sources with help of a well trained convolutional neural network.

6. Neural Network Datasets

We will use the Point of Gaze Eye-Tracking Dataset from the University of Texas at Arlington[2]. This provides a mapping from a video of eyes to locations in 3D space where the gaze of a user is focusing their attention. The dataset consists of data captured from 20 subjects. The dataset for each subject consists of a video of the users eye, transform matrices that relate the users eye and the screen of a display, and the target location on the screen that the subject is looking at. There are 6 scenarios for each of the 20 users. The transform and the target locations are provided per frame of the video. We also plan on using the comprehensive head pose and gaze database[4] to infer correlation between the pose of the head and the target that the subject is looking at.

We hope to run the system in a real-time manner and compare our results to expected locations on the display. In this setup data that we capture from the camera connected to the system will be used as our system inputs and data to process.

7. Progress

Despite some initial hurdles, we are still on track in terms of milestones we had initially set for ourselves. We are yet to build test framework to evaluate our solution. The test framework is conceptualized in the Evaluation section of this report.

7.1. Neural Network

We currently have the neural network implemented in TensorFlow, but has not been trained yet with our dataset.

7.2. Timeline

- 2/9/18 - Familiarize with opencv and setup base infrastructure for face/eye detection
- 2/16/18 - Face and Eye Detection
- 2/23/18 - Setup a neural network with existing database

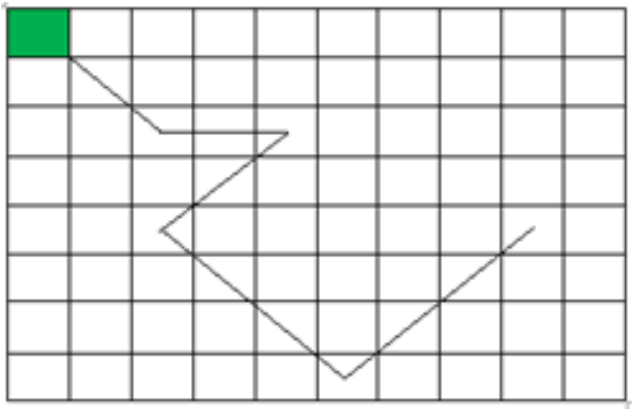

Figure 6. Path followed by Object on Screen.

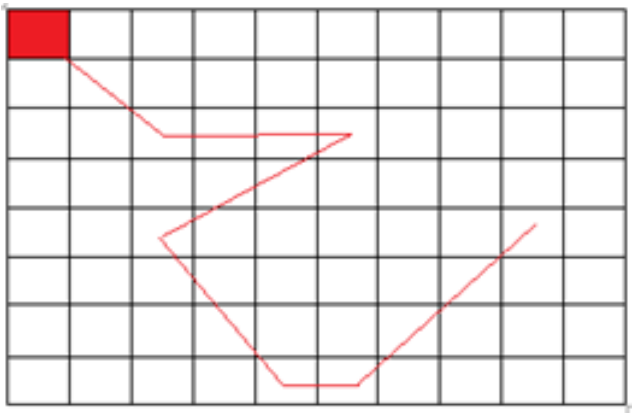

Figure 7. Path generated by tracking users eye.

- 2/26/18 - Compile and submit progress report
- 3/9/18 - Trained the neural network ready for inference
- 3/16/18 - Generate results and demo

8. Evaluation

We propose to evaluate our results by having the user follow an object around the screen on a known path (Figure 6). This would be our target. We can then compute the gaze of the user from the observations by the system (Figure 7). We can compute the error between the computed target from the gaze of the user and the known target location. From these two measurements we can compute the mean squared error between the target and the computed gaze location. We hope to achieve results that are within the size of the panels used for a keyboard on the screen. If the error is within the size of a key, then we can determine what keys a user is looking at, and even use the system as an input device.

References

[1] Calvo A. et al.. Eye Tracking Impact on Quality-of-Life of ALS Patients. In: Miesenberger K., Klaus J., Zagler W., Karshmer A. (eds) Computers Helping People with Special Needs. ICCHP 2008. Lecture Notes in Computer Science, vol 5105. Springer, Berlin, Heidelberg, 2008

[2] http://heracleia.uta.edu/ mcmurrough/eyetracking/

[3] Krafka, Kyle, et al. Eye tracking for everyone. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016

[4] McMurrough, Christopher D., et al. "A dataset for point of gaze detection using head poses and eye images." *Journal on Multimodal User Interfaces 7.3* (2013): 207-215.

[5] Weidenbacher, U.; Layher, G.; Strauss, P.-M.; Neumann, H.: "A comprehensive head pose and gaze database", *IET Conference Proceedings*, 2007

[6] Baluja, Shumeet, and Dean Pomerleau. "Non-intrusive gaze tracking using artificial neural networks." *Advances in Neural Information Processing Systems*. 1994.

[7] Cazzato, D., Dominio, F., Manduchi, R., and Castro, S. M., Real-time Gaze Estimation Via Pupil Center Tracking, Paladyn. *Journal of Behavioral Robotics*, In Press.

[8] Li, Tianxing, et al. "Ultra-Low Power Gaze Tracking for Virtual Reality." *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017.

[9] LeCun, Yann, et al. "Handwritten digit recognition with a back-propagation network." *Advances in neural information processing systems*. 1990.

[10] Le Cun, Yann, et al. "Handwritten zip code recognition with multilayer networks." *Pattern Recognition, 1990. Proceedings., 10th International Conference on*. Vol. 2. IEEE, 1990.

[11] http://www.eyetracking.com/Hardware/Eye-Tracker-List

[12] https://www.tobiidynavox.com/en-US/products/devices/

[13] https://www.tobii.com/group/about/this-is-eye-tracking/