Gaze Tracking

Nityam Dixit
Btech Computer Science
AU2040140
nityam.d@ahduni.edu.in

Devyash Shah
Btech Computer Science
AU2040152
devyash.s@ahduni.edu.in

Kavan Gondalia
Btech Computer Science
AU2040030
kavan.g@ahduni.edu.in

Kush Patel
Btech Computer Science
AU2040137
kush.p3@ahduni.edu.in

***Abstract*—The eye-tracking technology has shown continuous increase in its usage in mobile and its potential applications in various fields. The use of pervasive eye tracking can provide valuable information for a range of applications, such as life logging, user authentication, and mobile communication. We mention the two main types of eye-tracking technology, U-net based and pupil Model creaction,.**

***Index Terms*—Eye tracking, corneal reflection, pupil tracking, WebGazer, gaze tracking.**

I. Introduction

Eye tracking technology has evolved significantly in recent years, with attention shifting towards eye-based interaction in mobile and pervasive settings. Gaze behavior is a key indicator of human attention and provides rich information for understanding interactions with real-world environments. Pervasive eye tracking enables continuous monitoring and analysis of eye movements, offering potential applications in fields such as life logging, display interactions, user authentication, and mobile communication. This new paradigm of eye tracking requires an interdisciplinary approach at the intersection of ubiquitous computing, eye-tracking research, ego-centric vision, cognitive psychology, and design to address the implications and challenges of mobile eye-based interaction.

II. Literature Survey

According to Theory, Practice, and Standardization of Eye-tracking Technology (2018), there are two main types of eye-tracking technology: corneal reflection and pupil tracking. The former uses a camera and light source to track the reflection of light on the cornea, while the latter uses a camera to track the movement of the pupil. The latter method is the one you are using in your gaze tracker.

WebGazer is a popular open-source eye-tracking library that utilises machine learning to estimate a user's gaze location on a webpage (Papoutsaki et al., 2016). According to the LearnOpenCV video on Gaze Tracking and Estimation (2021), this library uses a simple algorithm to track the user's gaze by measuring the distance between the pupil and the eye corners. This can be useful for a variety of applications, including website design and marketing research.

Another implementation proposed by JOHANNES DESELAERS in Deep Learning Pupil Center Localization consists of 3 approaches. For all the models, Adam optimizer with a learning rate of 0.0002 and momentum values of 0.9 and 0.999 are used along with A mean square error (MSE) loss. ReLU is used as the activation function everywhere and a truncated normal distribution for weight initialization. One network uses dropout is used as a regularisation parameter. The first approach is a regression to 2D-coordinates using fully connected layers at the end of the network. The second approach is a Fully Convolutional Network that outputs a heatmap of probabilities for the location of the pupil center. The third approach involves pupil center localization as an extension of a semantic segmentation problem. After comparing through the models, the second method is found to be best for the dataset obtained from Tobii.

III. Implementation

Corneal reflections do not affect the method for recognising pupils in IR-lighted eye-camera pictures. The following is our standard pupil detection algorithm: The eye camera picture is first converted to grayscale. Followed by applying a blur filter, and then the starting area of the pupil is approximated using the most vital response centre-surround characteristic. After getting an image by the threshold, use a method to fit an ellipse.

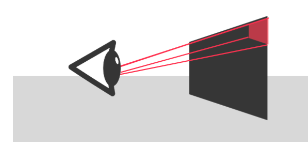


Fig. 1: Gaze detection on a screen

For more accurate and robust results that works better in noisy envoirnment, we have two approaches. To build the eye model, we have used a glint free method to detect the eye center in 3D geometry. It is very essential to have an accruate model to get an accruate estimate of the gaze direction and a model will help to generate and visualize the gaze direction.

A. U-Net model

The first approach is U-Net which is a widely used model for image segmentation which is based on contracting and expansive path. The former reduces the spatial dimensions of the input image and extracting features while the latter increases the spatial dimensions and reconstructs the image and both are connected using skip connections.
The loss function used is Binary Cross Entropy for training between the ground truth prediction and network prediction.

$$H(p,q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(q_i) + (1-y_i)\log(1-q_i)$$

where y_i is the truth value taking a value 0 or 1 of the i-th example, q_i is the predicted probability of the positive class, and N is the total number of examples.

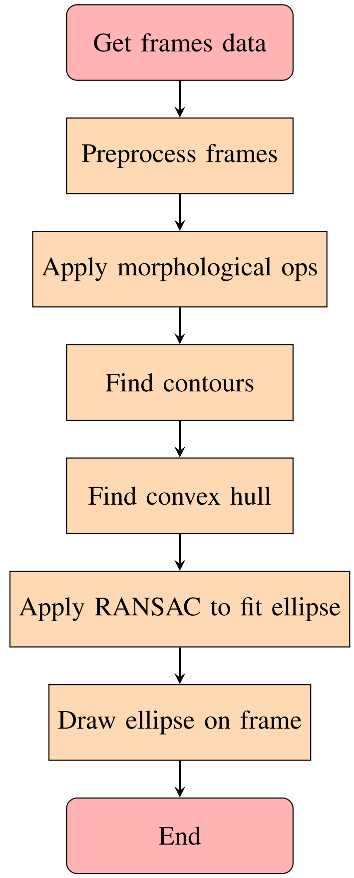
The input shape of the model is (480, 640, 3), and the output shape is (480, 640, 1). The model contains a series of convolutional layers, dropout layers, and max-pooling layers, followed by transposed convolutional layers to perform upsampling. To combine the feature maps from the contracting path with the corresponding upsampling feature maps from the expansive path a concatenation operation is used. In total the model has a total of 1,941,105 parameters, which are all trainable. The optimizer used is Adam, and the loss function used is binary cross-entropy with accuracy as the metric.

B. RANSAC(Random Sample Consensus)

The second approach used is RANSAC algorithm which works by iteratively selecting a random subset of data points (in this case, pupil candidates), fitting a model to those points (such as a circle or ellipse), and then evaluating the goodness of fit based on a predefined threshold. For our pupil detection, RANSAC is used to identify the best-fitting circle or ellipse among a set of pupil candidates, which is then selected as the final estimate of the pupil location.

For preprocessing, the frames are converted to grayscale, applied Gaussian blur and threshold them. The thresholded image is then processed using opening and closing morphological operations. This image then is used to find contours.

For each contour found, we use RANSAC algorithm to iteratively sample a subset of contour points, fit ellipse using least spuare methods and obtain ellipse parameters. These parameters would be used to see the detected pupil in the frame.



IV. Results

Both approaches performed well, U-Net is able to handle complex images while the RANSAC algorithm is Robust and more immune to outliers. Though with enough data, U-Net will perform with high accuracy. Our implementation resulted in high accuracy in both the approaches. U-Net however using more amount of computational power and hence is slow as it works on encoding and decoding the frame. Each frame of a video took approximately 180 to 200 milliseconds to process and thus resulted in a lower frame rate. RANSAC on the other hand was fast to compute and hence gave more smooth output with higher frame rate.

Below is the image representation

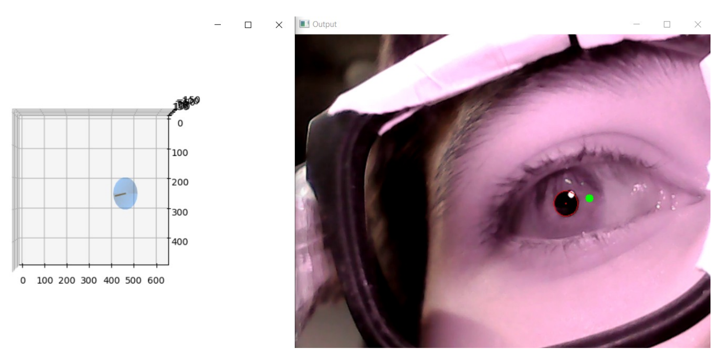


Fig. 3: Output of RANSAC Algorithm

V. Conclusion

To sum up, eye tracking technology has come a long way in recent years and has found many applications in mobile and pervasive settings. The ability to monitor and analyze eye movements continuously has opened up many possibilities for research and development. Pupil tracking is a popular method used in eye-tracking technology, and this article discussed an algorithm for detecting pupils in IR-lighted eye-camera pictures. By fitting an ellipse to the pupil movements, the accuracy of gaze estimation can be improved, and multiple calibrations are done to map eye movements to the real-world coordinate system. As this technology becomes more common, there is a need for interdisciplinary approaches to address the challenges and implications of mobile eye-based interaction.

References

[1] Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). Eye-tracking: A comprehensive guide to methods, paradigms, and measures. Oxford University Press.

[2] Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., & Torralba, A. (2016). WebGazer: Scalable webcam eye tracking using user interactions. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI) (pp. 3839-3845).

[3] Baltrusaitis, T., Robinson, P., & Morency, L. P. (2016). Real-time eye tracking and blink detection with USB cameras. Behavior research methods, 48(1), 265-274.

[4] Chauhan, R., Pujari, A. K., Raman, B., & Rajesh, R. (2018). Pupil detection for wearable eye-tracking using SIFT features. In 2018 IEEE 15th India Council International Conference (INDICON) (pp. 1-5). IEEE.

[5] @miscDESELAERS2020, author = Deselaers, Johannes, title = Deep Learning Pupil Center Localization, year = 2020, howpublished = https://www.diva-portal.org/smash/get/diva2:1509851/FULLTEXT01.pdf,