EYES OF THE TIGER

Furkan Ünal - Harun Büyüktepe - Ahmet Yasin Öztürk

SUPERVISORS:

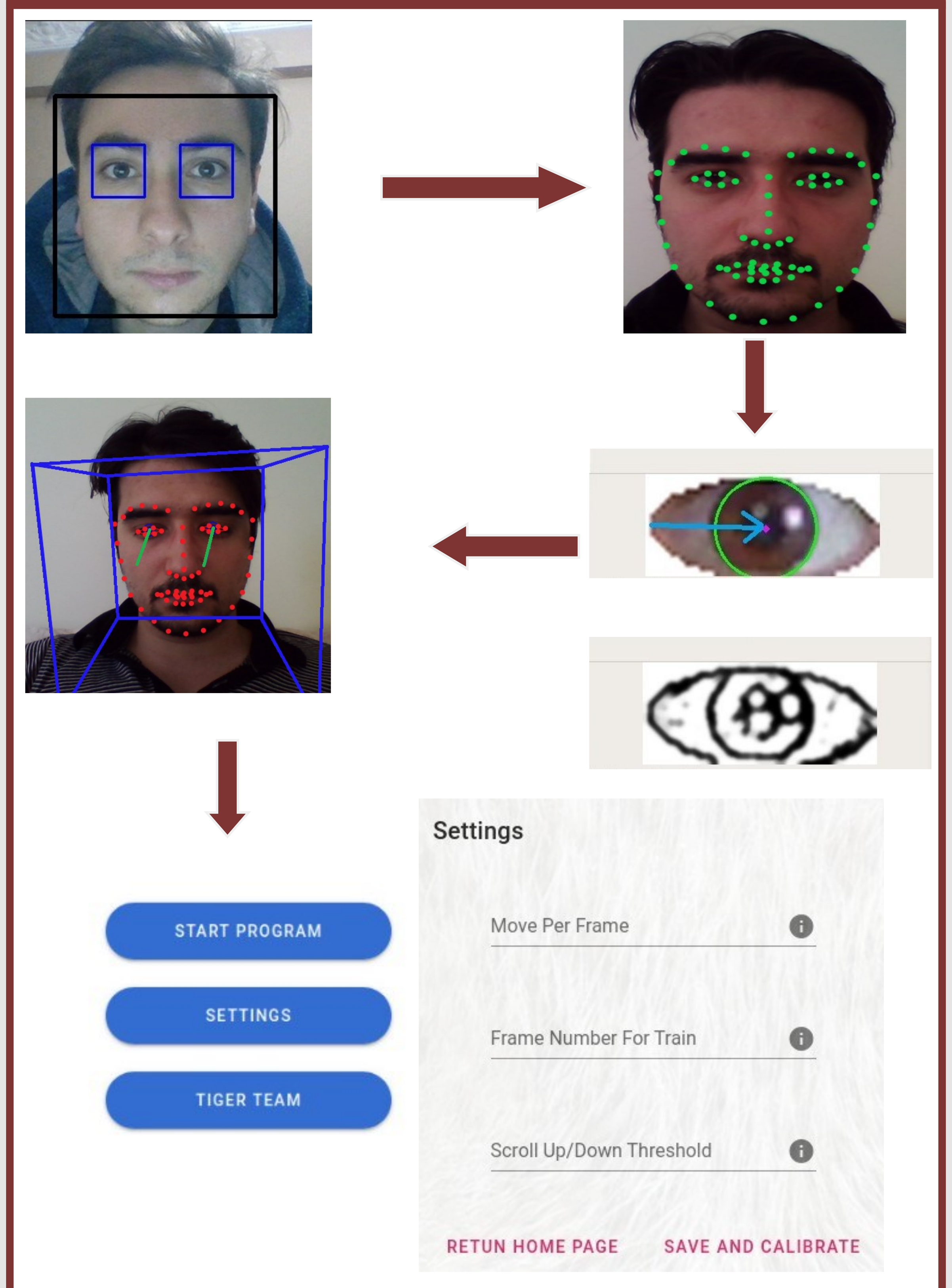
Assoc. Prof. Dr.
M. Borahan Tümer

Prof. Dr.
Çiğdem Eroğlu Erdem

INTRODUCTION

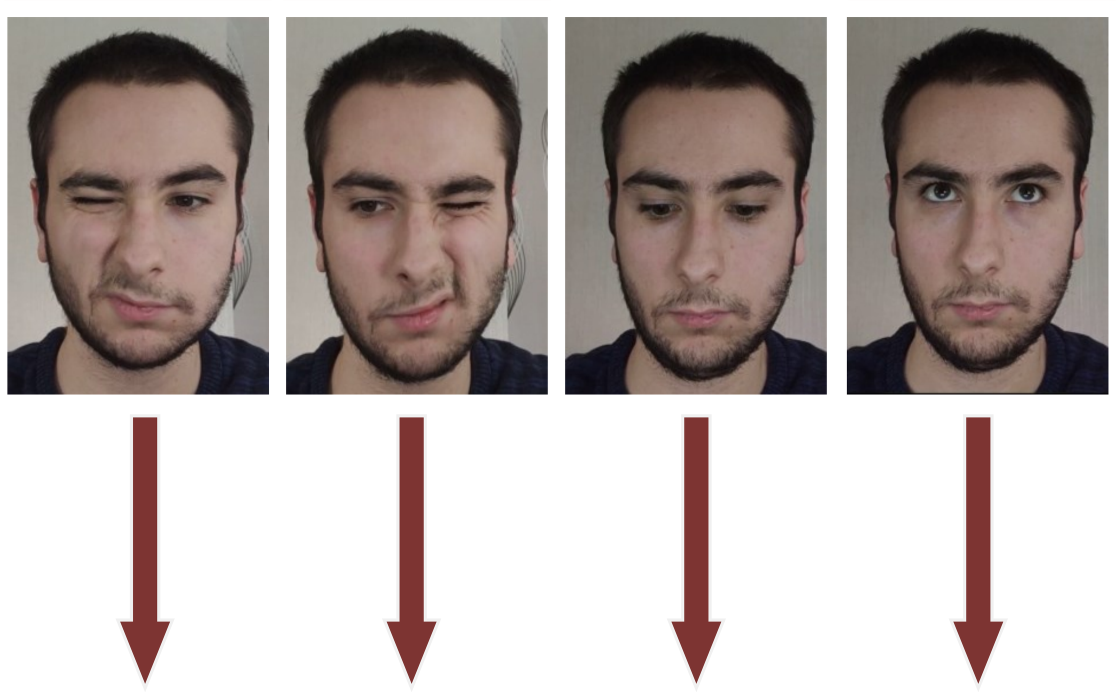
- Leaving the keyboard and taking control of the mouse is a time consuming job.
- For disabled people computer usage is impossible when they cannot use their hands freely.
- We designed an eye-gaze and blink controlled system using a webcam environment which will replace the mouse.

MAIN STEPS



MOUSE FUNCTIONALITY

Right Blink	Left Blink	Look Down	Look Up



Right Click	Left Click	Scroll Down	Scroll Up

REFERENCES

[1]Y. Cheung and Q. Peng, "Eye Gaze Tracking With a Web Camera in a Desktop Environment," in IEEE Transactions on Human-Machine Systems, vol. 45, no. 4, pp. 419-430, Aug. 2015.

[2]Soukupová, Tereza and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks." (2016).

[3] Fogelton, A., n.d. Eyeblink - Research. [online] Blinkingmatters.com. Available at: <https://www.blinkingmatters.com/research> [Accessed 7 June 2020].

[4]R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," IEEE Trans. Image Process., vol. 21, no. 2, pp. 802–815, Feb. 2012.

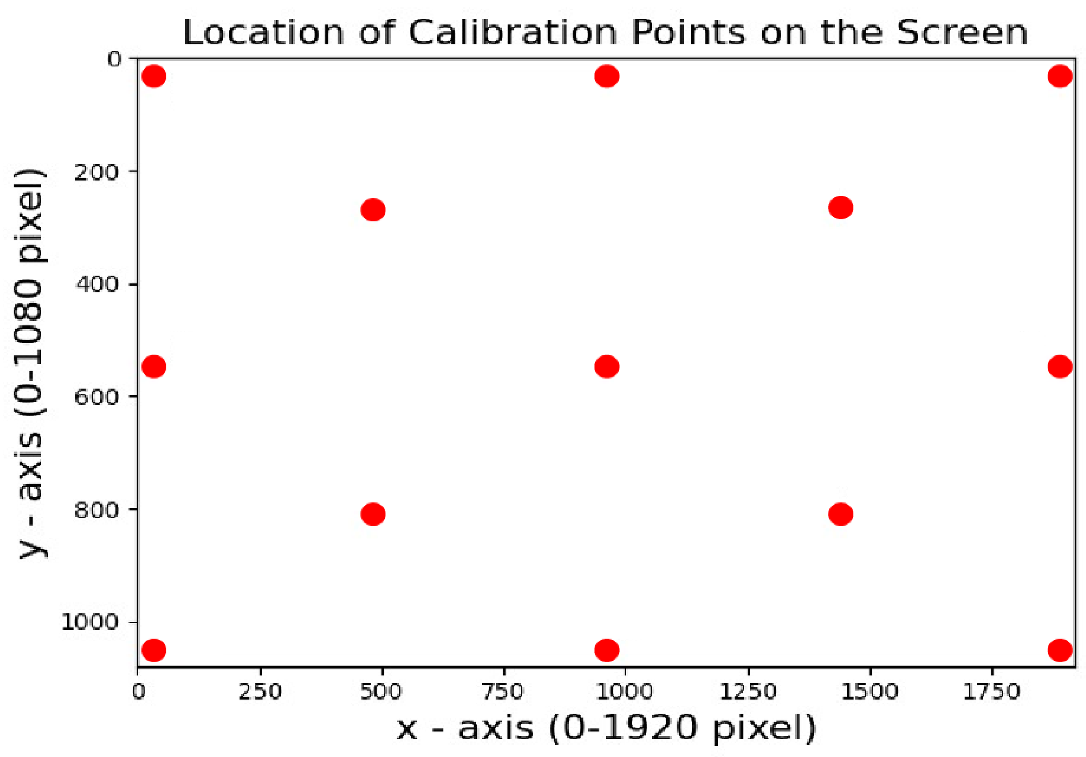
[5]B. Amos, B. Ludwiczuk, M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

MAPPING

Calibrate the system such that estimates the screen coordinate that user stares at.

Calibration :

Users are asked to stare at 13 different calibration points on the screen for 45 frames. System extracts necessary data to use estimating gaze point.



Mapping with Polynomial Regression [1]:

$[g_x, g_y]$: Eye Vector

$[u_x, u_y]$: Coordinates on the Screen

$[a_n, b_n]$: Polynomial Coefficients

$$u_x = a_0 + a_1 g_x + a_2 g_y + a_3 g_x g_y + a_4 g_x^2 + a_5 g_y^2$$
$$u_y = b_0 + b_1 g_x + b_2 g_y + b_3 g_x g_y + b_4 g_x^2 + b_5 g_y^2$$

Mapping with Support Vector Regression:

$[T_x, T_y, T_z, R_x, R_y, R_z]$: Head Pose

$[G_x, G_y]$: Gaze Angle

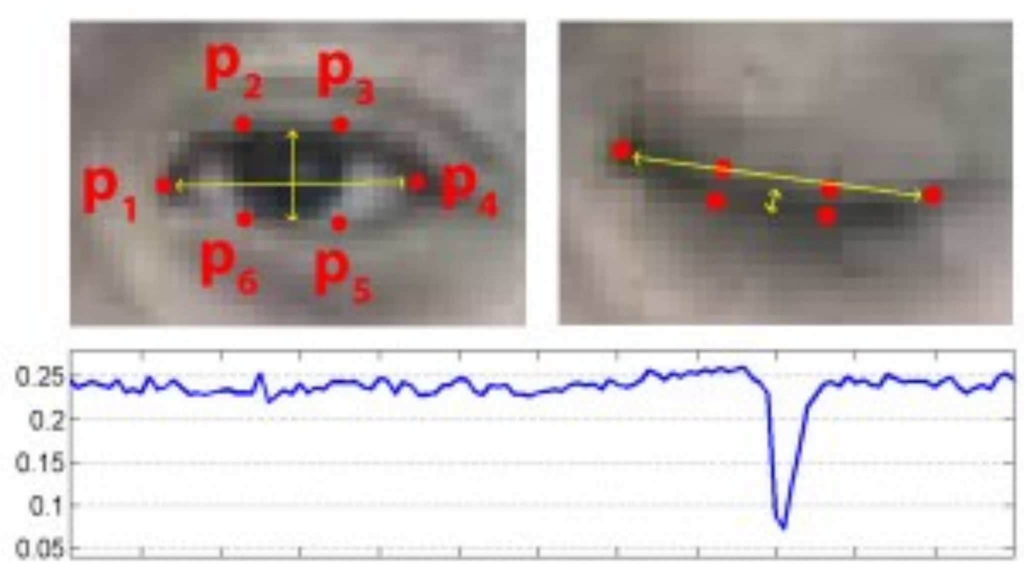
$[E_x, E_y, E_z]$: Left Eye Vector

$[E_x, E_y, E_z]$: Right Eye Vector

After scaling, data is used as input to the Support Vector Regression model for training and for prediction.

BLINK DETECTION

- Support Vector Machine model to predict blinks with eye aspect ratio (EAR) [2].
- Model trained with eyeblink8 dataset [3] and was tested with talkingface dataset [3].



Scores	Precision	Recall	F1-Score	# Samples
1	0.87	0.86	0.86	453
0	0.99	0.99	0.99	4535

Blink Test Scores

COMPARISON

Accuracy(Error in Degrees) Formula [1]:

$$A_{dg} = \arctan\left(\frac{A_d}{A_g}\right)$$

$[A_{dg}]$: The accuracy of the gaze tracking system in angular degree.

$[A_d]$: Denotes the distance between the estimated and actual gaze point.

$[A_g]$: Represents the distance between the subject and the screen plane.

Method	Error in Degrees	Extracted Frame #
Tiger_v1	5.643°	13689
Tiger_v2	1.898°	15755
Valenti [4]	2.00°	-
Cheung and Peng [1]	1.28°	-

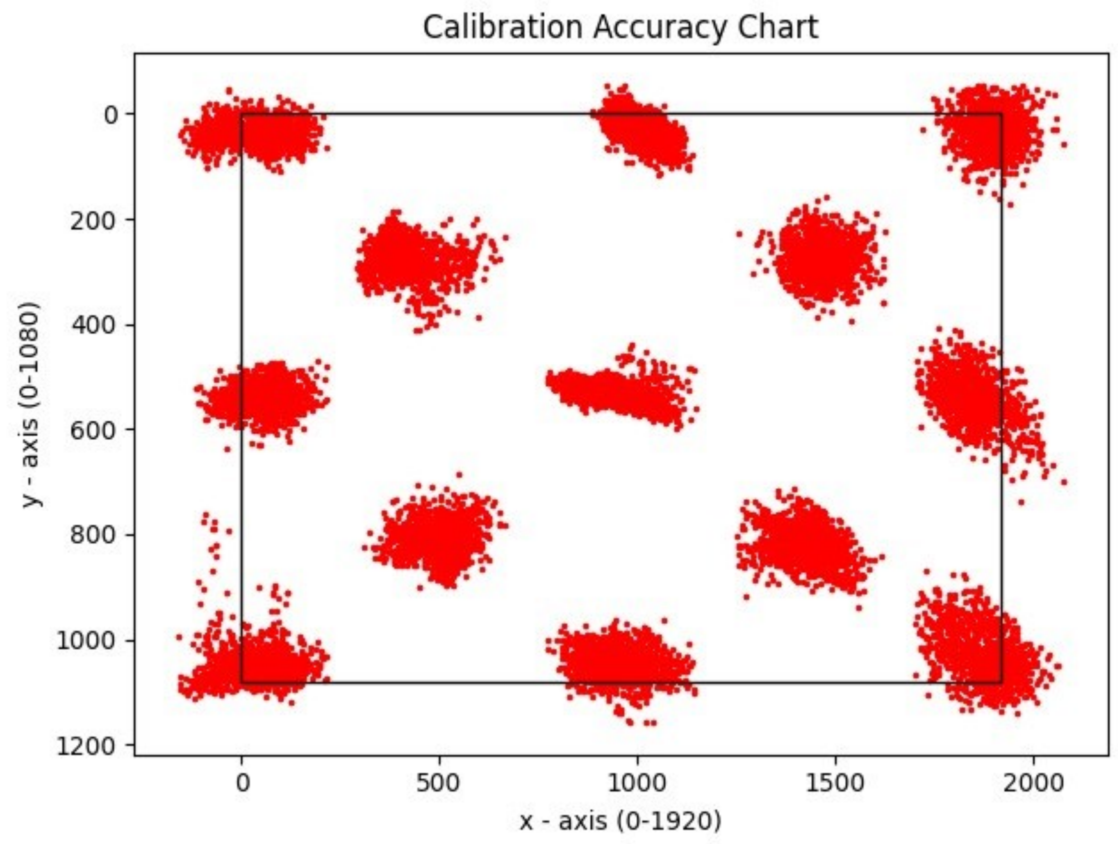
Comparison of Different Methods

We trained the Support Vector Regression model with different feature sets.

Used Features	Error in Degrees
HeadT, HeadR, GazeAngle	2.007°
* Gaze0, Gaze1	1.898°
* EV0, EV1	1.924°

*** : Contains previous rows features**

Support Vector Regression model accuracies with feature sets



Trained Model Predictions

CONCLUSION

- We built an eye-gaze and blink controlled system that works with only a webcam.
- In screen coordinate prediction using the Support Vector Regression with OpenFace [5] is better than the polynomial regression function with only using eye vectors.

TECHNOLOGIES USED

Python C++ Cmake Vue.Js Electron.Js Github