

# Eye-movement Analysis and Prediction using Deep Learning Techniques and Kalman Filter

Sameer Rafee<sup>1</sup>, Xu Yun<sup>2</sup>, Zhang Jian Xin<sup>3</sup>,  
School of Mechanical Engineering & Automation  
Zhejiang Sci-Tech University  
Hangzhou, China

Zaid Yemeni<sup>4</sup>  
Collage of Internet of Things (IoT) Engineering  
Hohai University  
Changzhou, China

**Abstract**—Eye movement analysis has gained significant attention from the eye-tracking research community, particularly for real-time applications. Eye movement prediction is predominantly required for the improvement of sensor lag. The previously introduced eye-movement approaches focused on classifying eye movements into two categories: saccades and non-saccades. Although these approaches are practical and relatively simple, they confuse fixations and smooth pursuit by putting them up within the non-saccadic category. Moreover, Eye movement analysis has been integrated into different applications, including psychology, neuroscience, human attention analysis, industrial engineering, marketing, advertising, etc. This paper introduces a low-cost eye-movement analysis system using Convolutional Neural Network (CCN) techniques and the Kalman filter to estimate and analyze eye position. The experiment results reveal that the proposed system can accurately classify and predict eye movements and detect pupil position in frames, notwithstanding the face tracking and detection. Additionally, the obtained results revealed that the overall performance of the proposed system is more efficient and effective comparing to Recurrent Neural Network (RNN).

**Keywords**—Eye Movement Classification; Eye Movement Prediction; Convolutional Neural Network (CNN); Recurrent Neural Network (RNN)

## I. INTRODUCTION

For a few decades, eye movement analysis has attained significant attention in the Human-Computer Interaction community. Various researchers have proposed using different technologies and algorithms to perform an automatic analysis for the position and direction of human eyes, helping different applications perform some analytics. The research community of eye movement analysis is growing owing to its capabilities of facilitating various tasks. One of these most useful is to detect someone's interest by determining the eye movement position and pupil size at a certain time. The analysis of eye movement has been commonly used in a variety of applications and different research areas, including visual systems [1], [2], [3], neuroscience and psychology [4], healthcare and psycholinguistics [5], user experience and interaction [6], consumer research, professional performance, and marketing [7], clinical research, economy, and education [8], software engineering, and product design [9], virtual reality [10], and transportation [11].

One of the main significant features of this research is to provide a method that can be exploited to accurately and objectively record and analyze human visual behavior. It would be incredible to request people glance over the supermarket

aisles to remember how many minutes they looked at every item or what advertisements they noticed most [12], [13]. Analyzing human eye movements enables researchers to study the movements of a participant's eyes during various activities, which provides insightfulness into the cognitive processes and understanding the common behavior of human behavior. Besides, it can show different things, including the methods of social interaction and learning patterns. It also helps researchers to screen typical nervous growth and perceptual disabilities. Furthermore, analyzing eye movements helps people clarify their thoughts [14], [15].

Eye movement analysis is referred to as the process of determining either the gaze point or the eye movement relative to the head position. In other words, Eye movement analysis classifies eye positions and eye movements by providing informative details about such movements. Eye movement analysis systems are widely utilized in various research fields, such as psychology, visual system, marketing, psycholinguistics, product design, and human-computer interaction. They are also ever-increasingly exploited for assistive and rehabilitative applications. Besides, along with eye movement analysis, it is required to estimate a noise-reduced signal's open eye movement position. Therefore, signal smoothing is obligatory to minimize the eye tracker's limited accuracy, and prediction is desirable to reduce the sampling lag of tracking systems. The eye movement prediction is meaningfully connected to distributed systems where instantaneous data of eye movements are required to be propagated over the network—the best example of this is Collaborative Virtual Environment (CVE) [10], [16].

There are numerous fields in which eye movement analysis could be applied extensively. Here to mention a few:

- **Scientific Research:** Studying visual behavior helps research obtain significant insightfulness into nervous growth, learning patterns, and cognitive impairments or diseases, such as people with Parkinson, Alzheimer, schizophrenia, brain injury, depression, autism, etc. In the same way, eye movement analysis can identify dyslexia and other learning or reading difficulties [17][18].
- **Market Research:** Eye movement analysis provides detailed and impartial information about the behavior of consumers and helps carry out decision-making processes. It enables brand owners and market researchers to detect consumers' behavior when observing and

choosing a product. It enables them to figure out what the most products naturally attracted the customers and the most product they ignored. Unlike questionnaires or surveys, eye movement analysis provides detailed, trustworthy behavior critical for branding, advertising, and packaging [16].

- Human Performance: Eye movement analysis and prediction can provide good insight into the ways of performing tasks and implementing processes implemented. It is used to identify risks, streamline training, operational inefficiencies, time efficiencies, and productivity improvement. Likewise, in sports and coaching, eye movement analysis can be used to improve performance by identifying skills and setting up strategies. It gives a unique insight into actions and tasks carried out hurriedly and often unconscious [19].

This paper introduces an eye-movement analysis and prediction system using a combination of the Kalman Filter and a convolutional neural network Network (CCN) to analyze the eye movement signal's instantaneous classification and predict forthcoming eye movement positions. Furthermore, using a numerical interpolation of the eye movement positions, the Kalman filter is utilized to provide a more useful interaction between the proposed system and the end-user. The contributions of this paper are as follows:

- Developing a real-time eye-movement analysis system using deep learning techniques, such as Convolutional Neural Network (CNN) and Kalman filter, which provides an accurate estimation for eye movements.
- Developing a cost-efficient eye-movement analysis and prediction system, which uses easy powerful deep learning-based calibration technique to predict eye movement and predict the eye movements based on the output of the deep learning technique.
- Designing hardware modules which are inexpensive based on a web camera and Arduino board.
- Conducting a quantitative evaluation of the performance for the implemented eye-movement analysis and prediction system.

The remainder of this paper is organized as follows: Section II presents state-of-the-art approaches related to the proposed system. Section III introduces the architecture of the proposed system. Section IV elaborates the evaluation of the proposed system and discusses the obtained results. Finally, Section VI concludes this work.

## II. RELATED WORK

Video/image-based eye movement analysis is classified into eye detection, eye position interpretation, and eye-tracking. It is critical to determine the eye model parameters from the image data in eye detection and tracking. The identification of the eye model can be made based on the eye region intensity distribution, the shape of the iris and eye, the pose of the head, etc. The authors in [20] and [21] present an excellent review of state-of-the-art image/video-based eye movement detection and tracking methods. As the authors stated, the eye detection models are categorized into the following classes:

- Eye-feature-based model;
- Eye-shape-based model;
- Eye-appearance-based model;
- Hybrid model.

### A. Eye-shape-based Models

The approaches based on this model detect the eye by contouring the pupil and iris and the eyelids' shape. Various eye-movement tracking applications use an elliptical model to prototype the eye, iris, and pupil's external shape. In these models, the model-fitting and voting-based methods are employed to calculate the elliptical model parameters. The authors in [22] exploited a voting-based method to contour the iris as an ellipse. They then applied horizontal scanning for the eye region to detect the iris center. At the horizontal line center, the pupil is located. The authors in [23] used ellipse to model the shape of the iris. Then, eye detection is carried out using spatio-temporal information. Because of the area darkness of the iris, the direction of the vertical and horizontal gradients will be outward from the iris center. Accordingly, a proposed voting-based method is used to find the iris center. At each edge point, an extrapolated line is delineated in the opposite direction of the gradient lines. The iris center is determined by the area where the largest number of lines are passing. In [24], the authors proposed a thresholding and contouring method to measure the radius of the iris. Other authors in [25] also used an ellipse to model the iris. Their model is built using RANSAC optimization and Expectation Maximization. In [26], the authors proposed an alternative ellipse fitting algorithm. They suppose that the contour points of the iris lay on an ellipse, and the contour points are applied to determine the ellipse parameters by a least-square fitting method. To obtain the sub-pixel level accuracy, the authors also introduced a sub-pixel edge detection approach. In [27] eye shape-based algorithm is proposed. The proposed algorithm utilizes a weakly supervised eye landmarks detection algorithm along with object detection and recurrent learning modules. The proposed algorithm can augment training data effectively and our specific format data consist of supervised and weakly supervised samples.

### B. Feature-based Models

The main goal of feature-based models is to detect local features, such as the pupil, limbus, etc., for localizing and modeling the eye parameters [20]. The authors in [28] tried to determine the region between the eyes rather than locating the local features. Their method is known as the "between-the-eyes" region. The idea behind this method is that the region between the eyes is brighter than the two sides. As both eyes are located on this region's sides, it contains the forehead and nose. Therefore, it is easier to be tracked, and both eyes can be detected by localizing both sides. In [29], the authors used the color of the skin to determine the face region. After that, in the detected skin area, both eyes are localized using four Gabor wavelets (linear filtering) to localize the eyes and non-linear filtering to detect the corners of both eyes. A feature-based gaze patterns recognition method is proposed in [30]. The proposed method used an eye tracker to collect data. Furthermore, Long Short-Term Memory (LSTM) technique is employed in this work for eye movements recognition.

### C. Appearance-based Models

These models are image template-based models. However, these models are single image-based templates and limited by the head pose changes and eye movements. If the head pose changes, these models fail. Also, these models are limited if there is a change in rotations and scales. Therefore, they need many different training datasets of eye areas to collect various eye orientations and states, and head poses in various illumination environments. A classifier is then modeled based on the pixel information of the given training datasets. Several authors introduced modified versions of feature-based models presented in [29] and [31] for eye detection and feature extractions. In [32] a human-computer interaction model is proposed based on medical staff eye movement. The Appearance-Based model is used such that the medical staff used their eye movements to control the robot during the operations time.

### D. Hybrid Models

Hybrid models are based on the three models introduced above. The authors in [33] and [34] use the appearance-based model as the combination of appearance and shape-based models by [20]. They also combined expectation-maximization for accurate estimation of the head poses and particle filtering for tracking the iris movements [21]. They used the distribution of face color and eye region for eye detection and tracking. In [35], the authors used image color information for face region detection. A hybrid eye movement recognition method is proposed in [36]. The proposed method aims to solve the inaccurate positioning problem of the initial position of the shape model in the process of eyelid matching by using machine learning. In addition, this method developed the algorithm by combining the AK-EYE model based on the combination of the ASM algorithm and Kalman filtering to create a local feature model for each feature point.

Similarly, the authors in [28] and [37] employed the same approach. After the face region detection, the detected face region is partitioned into smaller regions to find the eyes' region using thresholding and edge information. The template matching is then used for eye tracking. After the skin region detection, the detection of the face region is modeled as a face mirror-symmetry and ellipse. Besides, infrared illumination is also used for eye tracking, using more than one camera to capture IR images. If the light source is near the camera's optical axis, the light reflection from the pupil is used to detect the eye region. Accordingly, the pupil will be a brighter area in the image. Conversely, if the light source is far from the camera's optical axis, the pupil will appear as a darker area.

Different techniques for eye movement locating and measuring were proposed. Eye movement analysis is the process of locating the eye and measuring the eye movements. There are various methods and techniques used for eye-movement analysis. These techniques are divided into four classes:

- Scleral Search Coil (SSC)
- Infrared Oculography (IOG)
- Electrooculography (EOG)
- Video Oculography (VOG)

### E. Scleral Search Coil Technique (SSC)

As stated in [38], the SSC and Videoculography (VOG) techniques are primarily used for 3D eye-movement measurement. The authors proposed a method aiming to compare the accuracy of SSCS and VOG. Accordingly, the obtained results showed that VOG showed better accuracy than SSCS. Another Scleral Search Coil System was presented in [39]. The main goal is to investigate the effects of the Scleral Search Coils placement on the eyes saccades' kinematics. To that end, saccades of the human eye were recorded with an infrared video system while wearing coils. The results were compared with results while no coils were worn. According to [40], the SSC technique is suitable for animal studies, where coil lenses can be surgically implanted, leaving no peripheral to interfere with or damage. However, the SSC technique has several limitations. It can only be worn for a short period (approximately 30 minutes), and the eye cornea is required to be anesthetized. Even though it usually causes mild discomfort throughout and after the experimental sessions. The mounted annulus results in corneal dryness and vision blurring. The accuracy can be degraded by the annulus slippage on the eyes.

### F. Infrared Oculography (IOG)

This technique relies on the strength of infrared light reflected sclera to detect and localize the eye position using light beams generated by a pair of glasses. In this technique, a reference point, known as a glint or corneal reflection, is involved using an infrared light source when solving the sensitivity issue of head movements. It has been commonly used by various techniques, such as low pass filtering [41], K-means [42], Euclidean distance [43], random sample consensus [44], etc. The authors in [45] proposed a method based on a combination of IOG and Electromyography (EMG) techniques. The obtained result demonstrated that there is a robust linear correlation between time courses and amplitudes of EMG-and-IOG-recorded startle.

### G. Electrooculography (EOG)

As stated in [46], electrooculography (EOG) is an electrophysiologic test measuring the existing resting electrical potential between the Bruch's membrane and cornea. The mean transepithelial voltage of bovine Retinal pigment epithelium is 6 millivolts (mV). According to [47], the detection of saccades within raw mobile electrooculography (EOG) data includes complex algorithms processing data collected when seated static tasks. For dynamic tasks, the data processing is relatively infrequent and sophisticated, especially in the elders or people who have Parkinson's disease. EOG is an applicable and low-priced technique used in the field of human-computer interaction. This technique uses sensors attached to the eyes' surrounding area to obtain an electric field during eye rotation by measuring skin fluctuations. The eye movements are recorded unconnectedly employing electrodes.

Nevertheless, the recorded signal can be subjected to alteration without eye movements. Even if EOG is practicable, it is not widely used. To put it another way, the EOG is limited to medical applications and laboratories. Various approaches have employed this technique to perform different tasks [46][47]. The authors in [47] introduced a new method by employing a

differential electrooculography (EOG) signal using a fixation curve (DOSbFC) to eliminate baseline noise and drift based on a new electrode positioning scheme. The authors experimented with EOG eyeglasses and a new detection protocol for long-term step-by-step eye movement detection. The proposed DOSbFC computes the difference values of accumulated EOG signals between the initial eye movement and fixation time. It allows long-term eye movement detection with high accuracy and only needs a single calibration.

### H. Video Oculography (VOG)

The primary methods for three-dimensional eye movement measurement are the scleral search coil system (SSCS) and Video Oculography (VOG) [48]. The authors proposed a method to evaluate the accuracy of SSCS and VOG. The obtained results revealed that VOG recorded better accuracy than SSCS. Another scleral search coil system was presented in [49]. The proposed method investigates the extent to which scleral search coils' placement onto the eyes influences saccades' kinematics. The saccades were recorded using an infrared video system. Simultaneously, The coils were worn, and the main-sequence properties were compared with recordings without coils mounted on the eyes. According to [50], the VOG technique is the most widely used technique for currently eye-movement tracking systems.

## III. PROPOSED SYSTEM ARCHITECTURE

This section presents the general architecture and detailed components of the proposed system, which focuses on the analysis, classification, and prediction of eye movements. For the classification and prediction of eye movements, eye detection is the first step to be performed. The eye-detection step requires the head pose estimation for proper eye detection. Then, it is followed by finding the pupil to determine the iris center. The eye movement classification uses the positional eye data for the separation of eye movements: saccades, fixations, and smooth pursuit, which is a highly challenging task. Finally, the future eye movement prediction is carried out based on the classification of eye movements and head pose estimation.

### A. System Components

The main task of this paper is to construct a system for eye movement analysis and prediction. The system's practical aspect includes incorporating all the components given in detail in the following sections. Thus, the system comprises the following components, as shown in Fig. 1. As illustrated by Fig. 1, the proposed system comprises the following components:

- Hardware components
- Software components.

### B. Hardware Components

The hardware component is one of the essential parts of the proposed system. It is considered as the data frame source. The hardware component consists of sub-components: webcam and Arduino Uno board. In this section, the details of each sub-component are articulated. Fig. 2. shows a general connection view of the hardware components and the system. While Fig. 3 represents the real-world implementation of the system described in Fig. 2.

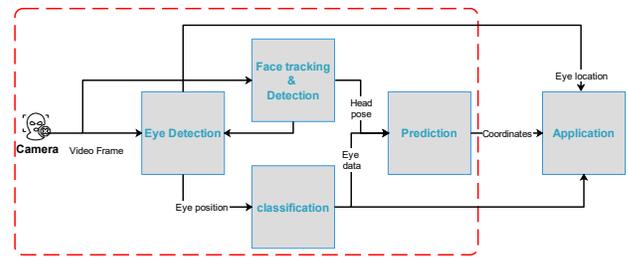


Fig. 1. The Proposed System's Architecture.

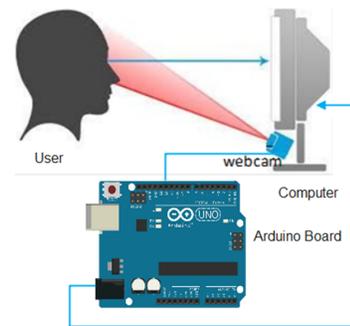


Fig. 2. Hardware Connection.

### C. Software Components

The software module is responsible for performing different tasks and consists of the following sub-modules:

- Eye Detection,
- Face Tracking & detection,
- Eye Movement classification, and
- Prediction.

The detailed procedure of each module is given below.

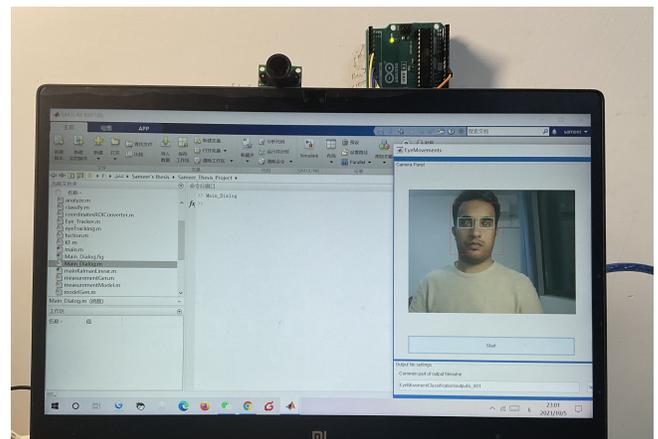


Fig. 3. Software Components.

1) *Eye Detection*: This component's main task is to detect the eyes on the face and determine the pupils' position. For eye detection, face region detection is the first task to be performed based on the head pose provided by the Head Pose Estimation module. The next step is to detect the eyes region and then detect the iris region.

a) *Eye Region Detection*: After face region detection, a feature-based method is applied sequentially to determine the regions of the left eye and the right eye. The gradient is calculated for each video frame based on the detected face box to accomplish this objective. Then, a horizontal projection is applied to calculate video frame gradient. In this application, the eye regions are located in the upper face part, characterized by the changeful value compared to the other face parts. After determining the horizontal position and the face dimensions, a vertical projection is applied to each part of the horizontal position. Owing to the area's brightness between the left and right eyes, the peak of the vertical projection is considered as the face center. Based on the horizontal and vertical projections, two peaks are located as the left and right boundaries of the video frame, and the face width is calculated based on the two lines of the vertical projection. Based on the face segmentations resulted from both vertical and horizontal projections, the widths and heights of both eye regions can be estimated. And then, the regions of interest (ROIs) can be calculated.

b) *Pupil Centers Localization*: To localize the pupils centers, an eye template must be created to match the eye size on the template and the size of the actual eyes on the video frame. Assume that the two eye regions' created template is denoted as  $T[i, j]$ , and  $V[i, j]$  represents the size of the eyes on the video frame, which is required to be detected. To calculate  $V[i, j]$ , the template can be placed on the video frame by calculating the template intensity values and the corresponding points of the image. Since the template does not absolutely match the video frame, the dissimilarity between the template intensity values with the corresponding intensity values of the video frame is calculated, as follows:

$$\max_{[i,j] \in TR} |I - V|, \sum_{[i,j] \in TR} |I - V|, \text{ or } \sum_{[i,j] \in TR} (I - V)^2 \quad (1)$$

where TR is the template region.

If there is a match between the template and the video frame. The dissimilarity measure is calculated by Equation (2).

$$\sum_{[i,j] \in TR} (V - T)^2 = \sum_{[i,j] \in TR} V^2 + \sum_{[i,j] \in TR} T^2 - 2 \sum_{[i,j] \in TR} (V \times T)^2 \quad (2)$$

Reasonably, this calculation for determining the template instances is to create a general template that can be used for localizing the pupils' centers of the other video frames. This is to decrease the computational overhead of repeating the calculation of pupils' centers each time. If we have a template of  $M \times N$  dimensions, Equation (3) is used, which is known as cross-correlation.

$$M[i, j] = \sum_{a=1}^M \sum_{b=1}^N T[a, b] \times V[i + a, j + b] \quad (3)$$

where  $a$  and  $b$  are the offset between the video frame and the template.

The equations mentioned above are used to calculate the cross-correlation when  $V$  and  $T$  are fixed. However, in the case of  $T$  is fixed, while  $V$  is varying,  $M$  will depend on  $V$ . Therefore, the normalization of the cross-correlation is carried out to solve this problem, as shown in Equation (4) and (5).

$$corr[i, j] = \sum_{a=1}^M \sum_{b=1}^N T[a, b] \times V[i + a, j + b] \quad (4)$$

$$m[i, j] = \frac{corr_{TV}[i, j]}{\sqrt{\left\{ \sum_{a=1}^M \sum_{b=1}^N V^2[i + a, j + b] \right\}}} \quad (5)$$

2) *Eye Movement Classification*: Eye movements are classified into three categories: fixation, saccades, or smooth pursuit. The eye regions obtained from the previous stage are used in a multiclass classification for predicting the eye movement on the next stage.

a) *Classification Threshold Setting*: To classify the eye movements, the proposed system relies on velocity thresholds of eye gaze. It adopts an adaptive threshold for the saccade classification. The saccades are initialized by a user-defined threshold of the time series velocity to a certain value. Based on the specified initial value (TV1), the new threshold is adaptively determined by calculating the variance of sub-thresholds (SV), as given by Equation (6).

$$TV_n = \overline{SV}_{n-1} + St.dev \times \sqrt{\frac{\sum (SV_{n-1} - \overline{SV}_{n-1})^2}{N - 1}} \quad (6)$$

where St.dev represents the number of standard deviations higher than the average velocity.

This step is repeated until a threshold velocity is stabilized, as given by Equation (7).

$$|TV_n - TV_{n-1}| < 1^\circ / \text{sec} \quad (7)$$

The previous steps are carried out to determine the saccades classification. Regarding pursuit classification thresholding, if the velocity exceeds the minimum velocity threshold (pursuit threshold), it is classified as a pursuit. Otherwise, it is classified as a fixation.

b) *Convolutional Neural Network (CNN)*: CNN is known as a feed-forward neural network used for different machine learning purposes. Although one of the drawbacks of CNN is the huge time needed for training, its output accuracy compared to other deep learning models is better. Our proposed system adopts a CNN model by specifying three convolution stages as shown in Fig. 4:

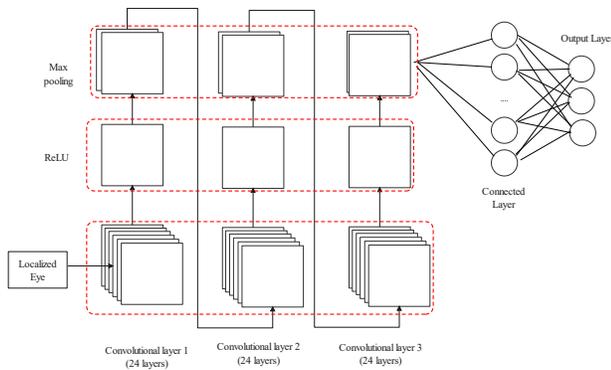


Fig. 4. The CNN Network Architecture.

- The input stage: the video frame is set to have a dimension of  $42 \times 50$ . The first layer utilizes 24 filters, and each filter is of  $7 \times 7$  dimensions.
- A rectified linear unit (ReLU) stage: In this stage, the ReLU layer is added, and a non-linearity to the activations is introduced, which is mathematically represented as given by Equation (8).

$$f(A) = \max(0, A) \quad (8)$$

where  $A$  is the input, and  $f(A)$  is the output of the ReLU layer.

- A max-pooling stage: In this stage, A max-pooling is followed the ReLU layer. It carries out a spatial sub-sampling of video frames. In this stage  $2 \times 2$  dimensional max-pooling layers have been used for the reduction of the spatial resolution of the video frames to half.

Moreover, two layers consisting of input, ReLU, and max-pooling layers are added after the previous stages with  $5 \times 5$  and  $3 \times 3$  dimensions, respectively. Then, a connected layer receives the output from the previous activation layer. In our case, the output layer consists of three nodes representing the eye movements: saccade, fixation, and smooth pursuit. The softmax loss function is used to measure the error, and the Cross entropy loss function is used as a loss function, which is reduced during the training phase. Equation (9) gives a mathematical representation of the Cross entropy loss function ( $L$ ).

$$L = -[t \log(f(A)) + (1 - t) \log(1 - f(A))] \quad (9)$$

where  $A$  is the classification vector,  $t$  is the truth value taking a value of 0 or 1, and  $f(A)$  is the softmax probability of  $i^{th}$  class

3) *Face Tracking and Detection*: This section is dedicated to elaborating on our solution for face tracking and detection. Fig. 5 presents an illustrative demonstration of the proposed solution for face tracking and detection.

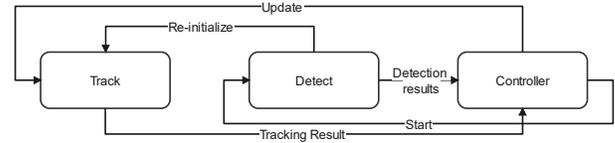


Fig. 5. Face Tracking and Detection Flowchart.

a) *Face Tracking*: Our proposed solution utilized a kernelized correlation filter (KCF) introduced in [51] for face tracking. Although there is a drawback of KCF when the face is not in the camera's view, it fails and hardly recovered. This challenge is solved using the detection unit, whose results are used to re-initialize the KCF algorithm and update the model with a specific learning rate. The procedure of KCF is summarized in the following steps:

- KCF uses a window that is 2.5 times larger than the given face to cycle shift to obtain training samples. Then, the extracted samples are labeled using Gaussian distribution and are saved in the filter model.
- KCF uses windows, which are 2.5 times larger than the best-obtained result to cycle shift to obtain the samples in the next frames. The similarity is calculated between the target in the KCF model and the samples. Then, the KCF is updated using the obtained result.

b) *Face detection*: For face detection, we adopt the lightweight multi-task cascade Convolution Neural Network. The following phases briefly describe the overall steps of face detection.

- Phase one: In this phase, a proposal Network is exploited to get the windows obtained from the face tracking unit and the related vector of their bounding boxes. Then. The regression vectors of bounding boxes are used to carry out the calibration of the candidates. The highly overlapping candidates are merged using non-maximum suppression.
- Phase two: The candidates obtained from phase one are used as an input to Refined Network to perform false candidate rejection. The non-rejected candidates with their boxes are calibrated using non-maximum suppression.
- Phase three: In this phase, the outputs of phase two are fed to O-net to obtain more accurate results. In this phase, we ignored the landmarks produced by O-net since we only need bounding surrounding the face to be used by the eye detection module.

The control unit controls the starting point of the face detection unit and updates the face tracking unit.

4) *Eye Movement Prediction*: This component's main task is to predict eye movement based on the output of head pose estimation and eye data of the classification component. This component works in conjunction with Head pose estimation, whose main task is to detect and estimate the pose direction in order to detect the eye position and direction. Eye movement

prediction can reliably track both eyes and detect the eye features in each video frame. However, by accurately detecting the number of head movements in each video frame, the user's eye movement estimation can be carried out accurately, as it is difficult for the users to keep their heads still.

In our proposed system, Kalman filter is modeled by two-state vectors to predict eye movements, as given by Equations (10) and (11), and the transitional state matrix is modeled as given by Equation (12).

$$x_n = \begin{bmatrix} \omega_x(n) \\ v_x(n) \end{bmatrix} \quad (10)$$

$$y_n = \begin{bmatrix} \omega_y(n) \\ v_y(n) \end{bmatrix} \quad (11)$$

$$A_n = \begin{bmatrix} 1 & \Delta s \\ 0 & 1 \end{bmatrix} \quad (12)$$

where  $\omega_x(n)$  and  $\omega_y(n)$  are the eyes' horizontal and vertical positions,  $v_x(n)$  and  $v_y(n)$  represent the horizontal and vertical velocities at time instance  $n$ , and  $\Delta s$  represents the sampling interval. The identity matrix for both state vectors is set as given by Equation (13).

$$H_n = [ 1 \quad 0 ] \quad (13)$$

To represent the noise resulted from Arduino board or the camera itself, The standard deviation of measurement noise  $R_n = \delta_v^2 = 1^\circ$ . Suppose there is a corruption in the eye signal position (the eye noise resulted from eye sub-movements, such as drifting). In that case, the estimation of system noise is done by the following covariance matrix, as given by Equation ((14)).

$$Q_n = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \quad (14)$$

where  $\sigma_w^2$  represents the variance of the system noise.

For eye movements predictions, the following is carried out:

- **Saccades prediction:** The prediction of saccadic movement is carried out using the method presented in [52]. To detect saccades, the differences between the actual eye velocity and the predicted eye velocity are measured. Two-State Kalman Filter is used to measure the predicted eye-velocity, and a chi-square test is employed to measure the difference between the actual eye velocity and the Kalman-filter-based predicted eye velocity, as follows:

$$V^2 = \sum_{i=1}^p \frac{(\hat{v}_2^-(i) - \frac{(z_i - z_{i-1})}{\Delta t})}{\sigma^2} \quad (15)$$

where  $\hat{v}_2^-(i)$  is the predicted velocity and  $z_i$  is the measured coordinate of eye position.  $\Delta t$  is the sampling interval and  $\sigma^2$  is the standard deviation of the actual eye velocity during  $\Delta t$ . We adopted the proposed method in [53] for determining the saccade

amplitude using the chi-square value. The proposed method was based on the following formula:

$$Sac\_amp = -0.000024x^6 + 0.0536x^4 + 1.5 \quad (16)$$

$$Sac\_dur = (2.2 \times Sac\_amp\_21)/1000 \quad (17)$$

Once  $Sac\_amp$  is calculated, the saccade duration (in seconds) can be calculated by Equation (17).

- **Fixation prediction:** fixation analysis is carried on by updating eye positions. The fixation is predicted when the eye velocity threshold is exceeding  $0.5^\circ$  per second for a minimum of 100 ms.
- **Smooth prediction:** this eye movement is predicted when the eye position does not belong to fixation or saccade eye movements, and the velocity is not exceeding  $140^\circ$  per second.

#### IV. SYSTEM IMPLEMENTATION & EVALUATION

When examining the proposed system, the realistic implementation environment is essential because it is exploratory by nature. This section is devoted to describing the implementation of the proposed system. It clarifies the environment in which the proposed system has been implemented and presents the prototype's functionalities.

##### A. Implementation Programming Environment

MatLab has been used for the proposed system implementation. This is because most of the proposed system computation is based on arrays (lists), which is simple to be implemented in Matlab. The version used for implementation was MatLab R2018b, which works on Windows 10 platform.

##### B. System Evaluation

For evaluating the proposed system, many experiments have been carried out to investigate the solution's success in different scenarios. Since the main objective of the proposed system is to classify and predict the movements, it has been evaluated in terms of eye movement classification and prediction. Different evaluation metrics have been employed, and an actual training dataset has been used. The following subsections are devoted to the description of the training dataset and evaluation metrics.

*1) Training Dataset Description:* The GazeCom dataset is the largest available dataset [54]. The dataset includes 18 recorded videos, and each video has 47 subjects. Far from the screen by 40 cm, the subjects were placed looking at the screen. The size of the screen was 40x30 cm with a resolution of 1280x720. The visual angle covered by the stimulus was 48x27 degrees, and each degree corresponds to 26.7 pixels. The recordings contain only monocular data; the resulted mean validation error was 0.62 degrees. The ARFF format was used to store the recordings due to the access simplicity by programming languages, such as Python and MatLab. The ARFF files contain different data fields, the time in microseconds, x and y positions, the scoring, and combined scoring. The eye movement classes contained in the dataset are unknown, saccades, fixations, smooth pursuit, and noise

numbered from 0 to 4, respectively. Other fields presented in the dataset are the direction, velocity, and signal acceleration. The whole dataset has been used to train the Eye Movement Classification module, which runs based on the CNN and RNN models. For testing, we used the data frame recorded by the system camera described above.

2) *Evaluation Metrics*: The following metrics are used to evaluate the performance of the proposed system in terms of eye movement classification and prediction. It is essential to specify quantitative and qualitative scores for the performance assessment of the proposed system and its counterparts in terms of eye movement classification, which can evaluate the proposed system in terms of eye movement prediction. The following evaluation metrics are selected.

- **Pursuits quantitative score (PQnS)**:  
The PQnS metric is used to determine the number of detected eye smooth pursuit, while the smooth pursuit of the stimuli is given. It is calculated by comparing the eye position  $(x_e, y_e, t)$  with the corresponding coordinate of smooth pursuit of stimuli  $(x_s, y_s, t)$ . If the eye position is classified as smooth pursuit, then the smooth pursuit detection counter increases. Equation (18) shows the mathematical representation of PQnS [51][55].

$$PQnS = \frac{\text{Saccade\_detection\_counter}}{\text{stimulus\_saccade\_points}} \times 100 \quad (18)$$

The ideal PQnS is calculated by Equation (19).

$$\text{Ideal\_PQnS} = \frac{n \times s_t + \sum_{j=1}^n D_{cor\_sac\_dur_j}}{\sum_{j=1}^n D_{stim\_pur\_dur_i}} \quad (19)$$

where  $n$  donates the pursuits of stimulus,  $D_{stim\_pur\_dur_i}$  represents pursuit duration stimulus  $i$ ,  $D_{cor\_sac\_dur_j}$  is the corrective saccade duration and  $s_t$  is the pursuit latency.

- **Fixation quantitative score (FQnS)**:  
This metric measures the number of detected fixation behavior to the number of presented fixations of stimuli. FQnS is calculated by sampling the fixation stimulus using a similar frequency of eye position. Then, the stimulus fixation coordinate  $(x_s, y_s)$  is compared with the recorded eye position coordinates  $(x_e, y_e)$ . The mathematical representation of FQnS is given by Equation (20) [51][55].

$$FQnS = \frac{\text{fixation\_detection\_counter}}{\text{stimulus\_stimuli\_points}} \times 100 \quad (20)$$

Where  $\text{fixation\_detection\_counter}$  denote the number of fixation points, when  $\text{stimulus\_stimuli\_points}$  the number of fixation stimuli is given.

- **Saccade quantitative score (SQnS)**:  
The main idea behind SQnS is to measure the number of the detected saccades when the stimuli saccadic behavior is given. The stimuli metric is calculated by considering each jump in the fixation target as a stimuli saccade, and distances between targets are added to the  $\text{total\_stimuli\_saccade\_amplitude}$ . Correspondingly,  $\text{total\_detected\_saccade\_amplitude}$  is the total of the absolute values of the detected saccade

amplitudes. The mathematical representation of SQnS is given by Equation (21) [51], [55].

$$SQnS = \frac{\text{total\_detected\_saccade}}{\text{total\_stimuli\_saccade}} \times 100 \quad (21)$$

- **Fixation qualitative score (FQIS)**:  
FQIS measures how the detected fixation is approximate to the given stimuli. Therefore, it provides the detected fixation's positional accuracy. Like FQnS, FQIS is calculated for fixation coordinates  $(x_s, y_s)$  of the given stimuli compared to the coordinates of eye position. if such coordinates are considered as a fixation, the distance between the centroid of the detected fixation coordinates  $(x_e, y_e)$  and given fixation coordinates is calculated. Equation (22) gives the mathematical representation of FQIS [51], [55].

$$FQIS = \frac{\sum_{i=1}^n \text{fixation\_distance}_i}{N} \quad (22)$$

$$\text{fixation\_distance}_i = \sqrt{(x_s^i - x_c^i)^2 + (y_s^i - y_c^i)^2} \quad (23)$$

where  $N$  is the number of stimuli position coordinates. The Ideal FQIS is  $0^\circ$ , which indicates the absolute accuracy of the system equipment.

- **Misclassified fixation score (MisFix)**:  
MisFix is calculated by separately measuring  $SP\_fixation\_points$ , the number of fixation coordinates in eye position classified as smooth pursuit, and the total number of fixation coordinates in stimuli ( $\text{total\_stimuli\_fixation\_points}$ ). Equation (24) gives the mathematical representation of MisFix [51][55].

$$MisFix = \frac{\text{sp\_fixation\_points}}{\text{total\_stimuli\_fixation\_points}} \quad (24)$$

The Ideal MisFix is 0%, meaning that no smooth pursuit was classified during the fixational stimulus.

- **Pursuit qualitative scores (PQIS)**:  
PQIS measures how the detected smooth pursuit is approximate to smooth pursuit in the given stimulus. PQIS is measured by  $PQIS\_P$  for position and the  $PQIS\_V$  for velocity accuracy. The  $PQIS\_P$  and  $PQIS\_V$  are calculated in a similar way to FQnS. The mathematical representations of  $PQIS\_P$  and  $PQIS\_V$  are given in Equations (25) and (26) [51], [55].

$$PQIS\_P = \frac{\sum_{i=1}^N \text{pursuit\_distance}_i}{N} \quad (25)$$

$$PQIS\_V = \frac{\sum_{i=1}^N \text{pursuit\_speed\_difference}_i}{N} \quad (26)$$

$$\text{pursuit\_distance}_i = \sqrt{(x_s^i - x_c^i)^2 + (y_s^i - y_c^i)^2} \quad (27)$$

$$\text{pursuit\_speed\_difference}_i = |v_s^i - v_c^i| \quad (28)$$

where  $N$  is the number of points of stimuli position. The evaluation ideal scores of this paper is shown in Table I.

TABLE I. THE EVALUATION IDEAL BEHAVIOR SCORES

SQnS	FQnS	PQnS	MisFix	FQIS	PQIS_P	PQIS_V
100	81.599	52.04	7.1	0	0	0

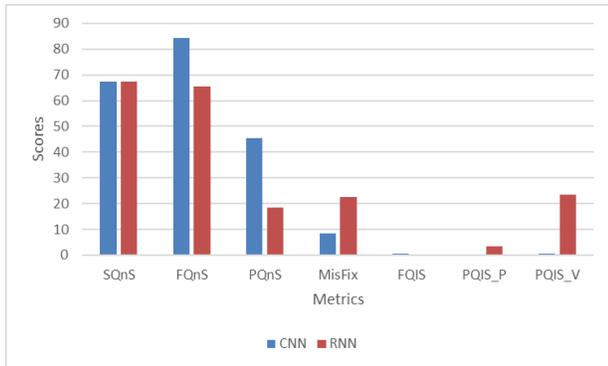


Fig. 6. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 30 Hz and Saccadic Threshold 75 Deg.

3) *Classification Evaluation:* In this section, the evaluation, in terms of eye movement classification, of the proposed system employing CNN against the RNN model is carried out based on the evaluation metrics given above. To evaluate the performance of the proposed system, the data frame was sampled using different frequencies (30 Hz, 100 Hz, 500 Hz, and 1000Hz).

The behavioral scores introduced in 4.2.2 were utilized to evaluate the effectiveness of the proposed system and its counterpart (RNN) in eye movement classifications.

Figs 6-9 display the obtained classification behavioral scores of both the CNN adopted in our proposed system and the RNN model, and the saccadic threshold is 75 deg. According to the obtained results for sampling rate 30 Hz, the CNN and RNN models achieve almost the same results for SQnS. However, the CNN model outperforms the RNN model in FQnS, PQnS, MisFix, FQIS, *PQIS\_P*, and *PQIS\_V*,

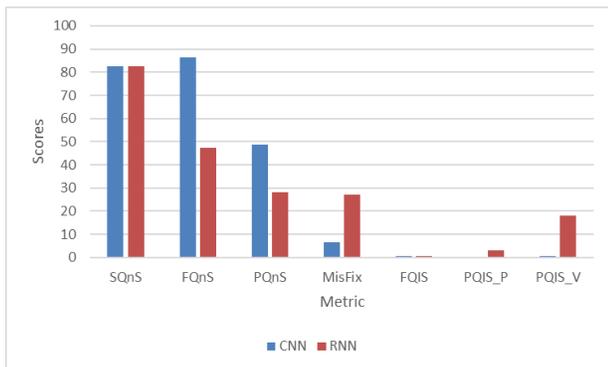


Fig. 7. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 100 Hz and Saccadic Threshold 75 Deg.

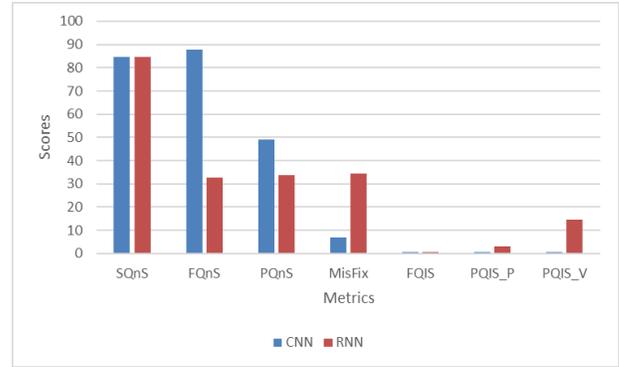


Fig. 8. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 500 Hz and Saccadic Threshold 75 Deg.

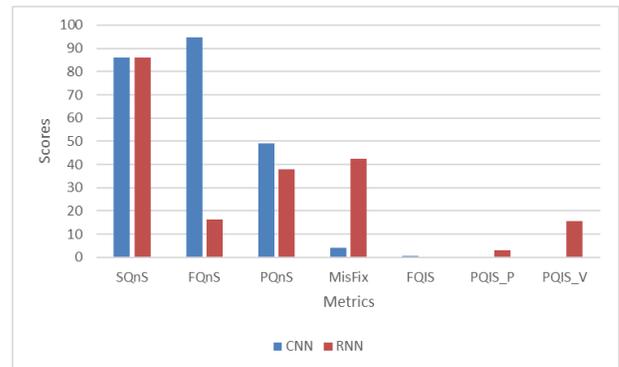


Fig. 9. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 1000 Hz and Saccadic Threshold 75 Deg.

respectively. For 100 Hz, it is noticeable the CNN model achieves better results than the RNN model, but are the close scores for saccades classification metric (SQnS). Similarly, for sampling rates (500 Hz and 1000 Hz), the CNN model shows better results than the RNN model. The RNN model shows an increase in MisFix as the value of the sampling rate increases. Besides, *PQIS\_P* and *PQIS\_V* show instability in measuring the pursuit position and velocity. It is worth mentioning that the CNN behavioral classification show proximity to ideal scores as the sampling rate frequency increases.

Moreover, the proposed system against the RNN model is evaluated in terms of eye movement classification with decreasing 10% of the saccadic threshold of the previous evaluation. As mentioned above, the evaluation is carried with different sampling rates, and the saccadic threshold is 65 degrees. We selected this slight change in the saccade threshold to determine the changes of eye movement classification that can take place with different classification thresholds of saccade movement. Fig. 10 to 13 depict the obtained classification behavioral scores of both the CNN adopted in our proposed system and the RNN model.

However, compared to the RNN model, the CNN model

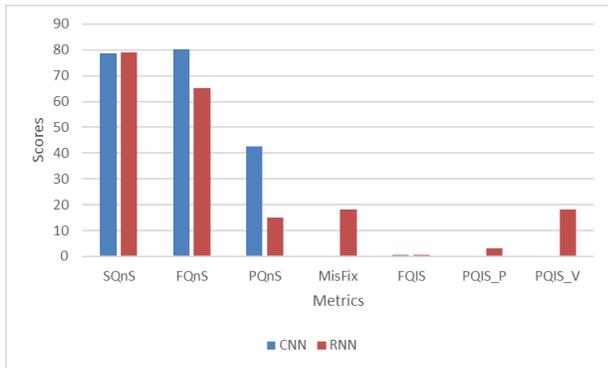


Fig. 10. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 30 Hz and Saccadic Threshold 65 Deg.

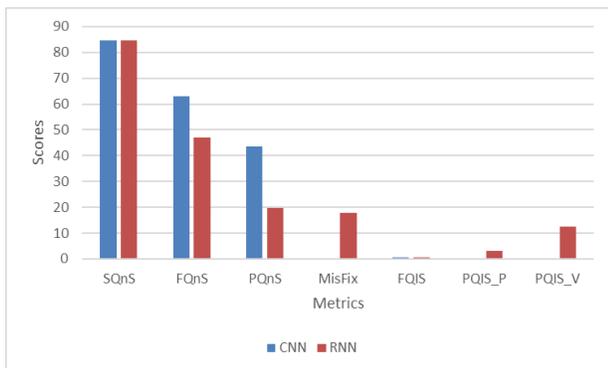


Fig. 11. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 100 Hz and Saccadic Threshold 65 Deg.

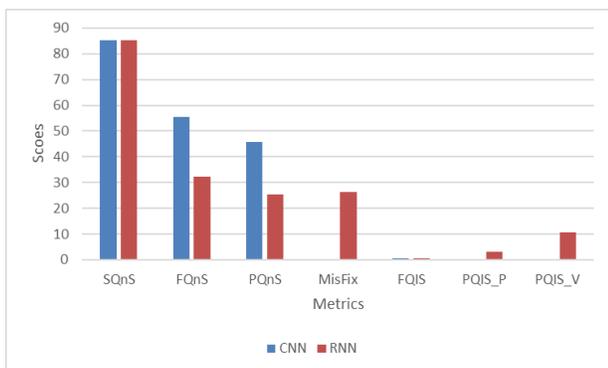


Fig. 12. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 500 Hz and Saccadic Threshold 65 Deg.

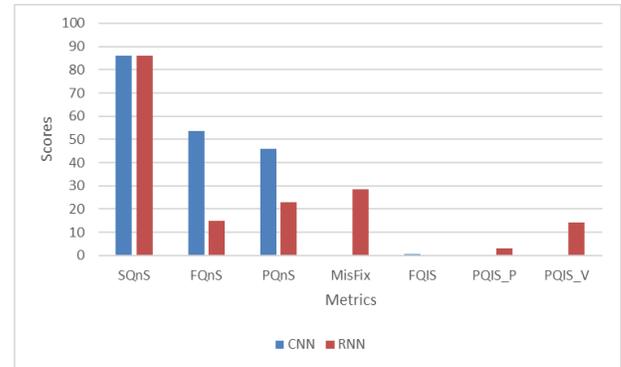


Fig. 13. The Behavioral Classification Scores of the Proposed System and RNN with Sampling Rate 1000 Hz and Saccadic Threshold 65 Deg.

achieves better results as indicated by the behavioral scores. Both models achieve almost the same SQnS scores for the given sampling frequencies. The other behavioral scores (FQnS, PQnS, FQIS, *PQIS\_P*, *PQIS\_V*) show that the achieved scores decrease as the sampling frequencies increase. This leads us to conclude that the saccadic threshold has effects on the behavioral classification scores as it decreases. It can be noticed that the CNN model outperforms the RNN model in achieving more reliable results. Table II gives the behavioral classification scores of the proposed system and RNN with various sampling rates, and the saccadic threshold is 75 deg.

Compared with the results listed in Table II, the eye movement classification performance of both the CNN and RNN models is notably decreased with the saccadic threshold decrease.

### C. Prediction Evaluation

The proposed system utilized the Kalman filter for eye-movement prediction with two states (velocity and position). The Kalman filter is applied to the recorded eye movement to predict the eye velocity and position. Then, the Chi-square test uses the measured and predicted position and velocity values to classify each positional value as fixation, saccade, and smooth pursuit. The eye movement prediction module of the proposed system is evaluated using two scenarios.

The first scenario was carried with a saccadic threshold of 75 degrees. The second scenario was carried out with a saccadic threshold of 65 degrees. Both scenarios were implemented with various sampling rate frequencies (30 Hz, 100 Hz, 500 Hz, and 1000 Hz). Figs 14-17 illustrate the behavioral scores of both scenarios.

As it can be seen, the different thresholds influence the Kalman filter prediction scores. The SQnS score has shown no changes in both scenarios for sampling frequencies (30 Hz, 100 Hz, and 500 Hz). As the sampling frequency increased to 1000Hz, the results have shown remarkable changes. For FQnS, it can be noticed that capability of the Kalman filter in classifying fixation is influenced by the increase in the sampling frequencies. This indicates that the accuracy of the

TABLE II. THE BEHAVIORAL CLASSIFICATION SCORES OF THE PROPOSED SYSTEM AND RNN WITH VARIOUS SAMPLING RATES AND SACCADIC THRESHOLD 75 DEG

sampling rates		SQnS	FQnS	PQnS	MisFix	FQIS	PQIS_P	PQIS_V
30 Hz	CNN	67.2945	84.3947	45.32	8.42	0.7092	0.2	0.5
	RNN	67.3928	65.4482	18.5856	22.4984	0.1772	3.3521	23.5821
100Hz	SQnS	FQnS	PQnS	MisFix	FQIS	PQIS_P	PQIS_V	
	CNN	82.495	86.4211	48.57	6.5	0.578	0.34	0.5
	RNN	82.6959	47.4095	28.0949	27.1789	0.5466	3.0109	17.9955
500Hz	SQnS	FQnS	PQnS	MisFix	FQIS	PQIS_P	PQIS_V	
	CNN	84.5859	87.7445	48.93	7.02	0.5543	0.7	0.5
	RNN	84.6112	32.7983	33.8999	34.3412	0.5508	3.1152	14.4503
1000Hz	SQnS	FQnS	PQnS	MisFix	FQIS	PQIS_P	PQIS_V	
	CNN	85.9343	94.6117	49.02	4.2	0.4984	0	0
	RNN	85.9343	16.2093	38.0532	42.3631	0.4505	3.1789	15.5107

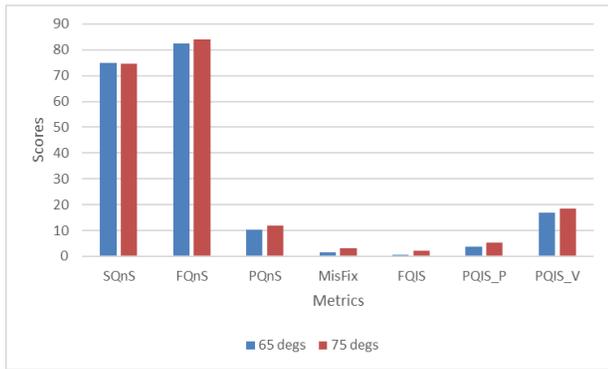


Fig. 14. The Behavioral Classification Scores with a Sampling Rate of 30 Hz and Saccadic Threshold 75 and 65 Degs.

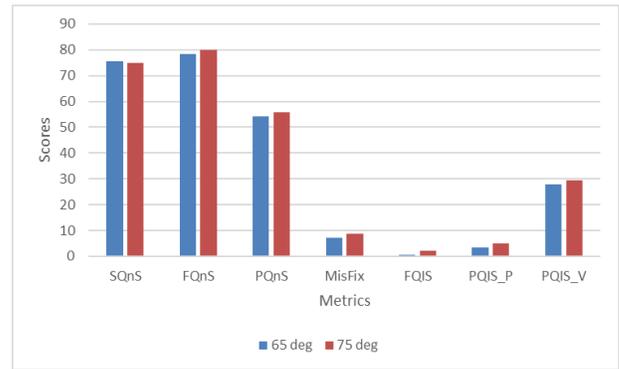


Fig. 16. The Behavioral Classification Scores with a Sampling Rate of 500 Hz and Saccadic Threshold 75 and 65 Degs.

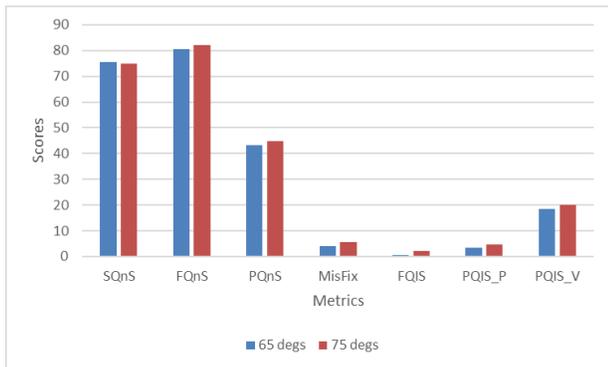


Fig. 15. The Behavioral Classification Scores with a Sampling Rate of 100 Hz and Saccadic Threshold 75 and 65 Degs.

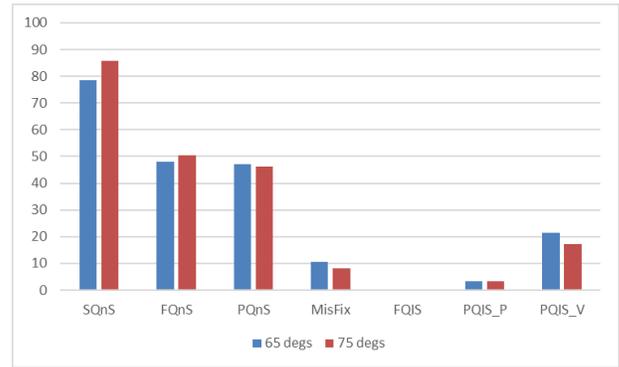


Fig. 17. The Behavioral Classification Scores with a Sampling Rate of 1000 Hz and Saccadic Threshold 75 and 65 Degs.

Kalman filter in fixation classification decreased as the sampling frequencies increase. Unlike FQnS, PQnS is increased in both scenarios. This indicates that the Kalman filter accuracy in pursuit classification as the sampling frequencies increase. However, the Kalman filter shows high fixation misclassification, as shown by MisFix scores. Moreover, the Kalman filter shows a deviation in predicting pursuit positional and velocity, as indicated by *PQIS\_P* and *PQIS\_V* scores, in both scenarios. This can happen in a case of a very high position detected before a very low position.

## V. CONCLUSION AND FUTURE WORK

### A. Conclusion

This paper proposed a system for eye movement analysis and prediction. The system proposed in this paper is a comprehensive system for three-type eye movement classification and prediction data consistency with a various and wide range of components. It combines two essential components: hardware and software. Each component encompasses different sub-components or modules, and each module has different func-

tions. The design of the proposed system relies on software designing architecture. We exploited one software architecture, namely, module architecture. As a result, the proposed system's design is highly flexible, and components can be accessed and used individually or collectively. Based on the obtained results, the proposed system accomplished good success in eye-movement classification and prediction.

### B. Future work

Eye movement classification and prediction is still an active research area. According to the work achieved in this paper, some research directions could be proposed.

- Empirical studies could cover a wide range of deep learning techniques to test and evaluate the behavior of the proposed system in eye-movement classification. Moreover, different lightweight techniques can be implemented to perform eye movement prediction.
- A recommended future research be done by employing different techniques for face tracking and detection to detect the face with different angles.
- Further recommended studies can be introduced to eye detection and localization. Besides, improvement of the developed prototype can be conducted. This includes the optimization of source Code and improvements of code speed, the implementation of the developed prototype in an implementation environment that support parallelization.

### REFERENCES

- [1] K. Harezlak and P. Kasproski, "Application of eye tracking in medicine: A survey, research issues and challenges," *Computerized Medical Imaging and Graphics*, vol. 65, pp. 176–190, 2018.
- [2] N. Hristozova, P. Ozimek, and J. P. Siebert, "Efficient egocentric visual perception combining eye-tracking, a software retina and deep learning," *arXiv preprint arXiv:1809.01633*, 2018.
- [3] B. H. Ulutas, N. F. Özkan, and R. Michalski, "Application of hidden markov models to eye tracking data analysis of visual quality inspection operations," *Central European Journal of Operations Research*, pp. 1–17, 2019.
- [4] M. Cutumisu, K.-L. Turgeon, T. Saiyera, S. Chuong, L. M. González Esparza, R. MacDonald, and V. Kokhan, "Eye tracking the feedback assigned to undergraduate students in a digital assessment game," *Frontiers in psychology*, vol. 10, p. 1931, 2019.
- [5] Z. CHEN, H. FU, W.-L. LO, and Z. CHI, "Artificial intelligence in medical applications," *Journal of Healthcare Engineering*, vol. 2018, p. 7692198, 2018.
- [6] K. Lukander, "A short review and primer on eye tracking in human computer interaction applications," *arXiv preprint arXiv:1609.07342*, 2016.
- [7] Y. Hang, X. Yi, and C. Xianglan, "Eye-tracking studies in visual marketing: Review and prospects," *Foreign Economics & Management*, vol. 40, no. 12, pp. 98–108, 2018.
- [8] T. Colliot and E. Jamet, "Understanding the effects of a teacher video on learning from a multimedia document: an eye-tracking study," *Educational Technology Research and Development*, vol. 66, no. 6, pp. 1415–1433, 2018.
- [9] U. Obaidallah, M. Al Haek, and P. C.-H. Cheng, "A survey on the usage of eye-tracking in computer programming," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–58, 2018.
- [10] R. B. Noland, M. D. Weiner, D. Gao, M. P. Cook, and A. Nelessen, "Eye-tracking technology, visual preference surveys, and urban design: preliminary evidence of an effective methodology," *Journal of Urbanism: International Research on Placemaking and Urban Sustainability*, vol. 10, no. 1, pp. 98–110, 2017.
- [11] V. Clay, P. König, and S. Koenig, "Eye tracking in virtual reality," *Journal of Eye Movement Research*, vol. 12, no. 1, 2019.
- [12] A. George, "Image based eye gaze tracking and its applications," *arXiv preprint arXiv:1907.04325*, 2019.
- [13] R. G. Lupu and F. Ungureanu, "A survey of eye tracking methods and applications," *Buletinul Institutului Politehnic din Iasi, Automatic Control and Computer Science Section*, vol. 3, no. 1, pp. 72–86, 2013.
- [14] H. Azami, S. E. Arnold, S. Sanei, Z. Chang, G. Sapiro, J. Escudero, and A. S. Gupta, "Multiscale fluctuation-based dispersion entropy and its applications to neurological diseases," *IEEE Access*, vol. 7, pp. 68718–68733, 2019.
- [15] Y. Terao, H. Fukuda, and O. Hikosaka, "What do eye movements tell us about patients with neurological disorders?—an introduction to saccade recording in the clinical setting—," *Proceedings of the Japan Academy, Series B*, vol. 93, no. 10, pp. 772–801, 2017.
- [16] K. Hercegfı, A. Komlódi, M. Köles, and S. Tóvölgyi, "Eye-tracking-based wizard-of-oz usability evaluation of an emotional display agent integrated to a virtual environment," *Acta Polytechnica Hungarica*, vol. 16, no. 2, pp. 145–162, 2019.
- [17] P. Beach and J. McConnel, "Eye tracking methodology for studying teacher learning: A review of the research," *International Journal of Research & Method in Education*, vol. 42, no. 5, pp. 485–501, 2019.
- [18] K. Harezlak and P. Kasproski, "Application of eye tracking in medicine: A survey, research issues and challenges," *Computerized Medical Imaging and Graphics*, vol. 65, pp. 176–190, 2018.
- [19] P. Bhatia, A. Khosla, and G. Singh, "A review on eye tracking technology," in *Interdisciplinary Approaches to Altering Neurodevelopmental Disorders*, pp. 107–130, IGI Global, 2020.
- [20] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 478–500, 2009.
- [21] D. W. Hansen and A. E. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.
- [22] K.-N. Kim and R. Ramakrishna, "Vision-based eye-gaze tracking for human computer interface," in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, vol. 2, pp. 324–329, IEEE, 1999.
- [23] R. Kothari and J. L. Mitchell, "Detection of eye locations in unconstrained visual images," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3, pp. 519–522, IEEE, 1996.
- [24] A. Pérez, M. L. Córdoba, A. Garcia, R. Méndez, M. Munoz, J. L. Pedraza, and F. Sanchez, "A precise eye-gaze detection and tracking system," 2003.
- [25] Q. Ji, "Special issue on eye detection and tracking," *Computer vision and image understanding (Print)*, vol. 98, no. 1, 2005.
- [26] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pp. 131–136, IEEE, 2002.
- [27] B. Huang, R. Chen, Q. Zhou, and W. Xu, "Eye landmarks detection via weakly supervised learning," *Pattern Recognition*, vol. 98, p. 107076, 2020.
- [28] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking eyes and monitoring eye gaze," in *Proc. Workshop on Perceptual User Interfaces*, pp. 98–100, 1997.
- [29] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1, pp. I–I, Ieee, 2001.
- [30] S. Yeamkuan and K. Chamnongthai, "Fixational feature-based gaze pattern recognition using long short-term memory," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1–4, IEEE, 2020.
- [31] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [32] P. Li, X. Hou, X. Duan, H. Yip, G. Song, and Y. Liu, "Appearance-based gaze estimator for natural interaction control of surgical robots," *IEEE Access*, vol. 7, pp. 25095–25110, 2019.

- [33] D. W. Hansen, D. J. MacKay, J. P. Hansen, and M. Nielsen, "Eye tracking off the shelf," in *Proceedings of the 2004 symposium on Eye tracking research & applications*, pp. 58–58, 2004.
- [34] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, and M. B. Stegmann, "Eye typing using markov and active appearance models," in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pp. 132–136, IEEE, 2002.
- [35] W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan, "Driver fatigue detection based on eye tracking and dynamic template matching," in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 1, pp. 7–12, IEEE, 2004.
- [36] S. Zhao, J. Luo, and S. Wei, "A hybrid eye movement feature recognition of classroom students based on machine learning," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2803–2813, 2021.
- [37] S. Tsekeridou and I. Pitas, "Facial feature extraction in frontal views using biometric analogies," in *9th European Signal Processing Conference (EUSIPCO 1998)*, pp. 1–4, IEEE, 1998.
- [38] N. Tsumura, C. Endo, H. Haneishi, and Y. Miyake, "Image compression and decompression based on gazing area," in *Human Vision and Electronic Imaging*, vol. 2657, pp. 361–367, International Society for Optics and Photonics, 1996.
- [39] K. Eibenberger, B. Eibenberger, D. C. Roberts, T. Haslwanter, and J. P. Carey, "A novel and inexpensive digital system for eye movement recordings using magnetic scleral search coils," *Medical & biological engineering & computing*, vol. 54, no. 2-3, pp. 421–430, 2016.
- [40] P. J. Murphy, A. L. Duncan, A. J. Glennie, and P. C. Knox, "The effect of scleral search coil lens wear on the eye," *British journal of ophthalmology*, vol. 85, no. 3, pp. 332–335, 2001.
- [41] M. Frens and J. Van der Geest, "Scleral search coils influence saccade dynamics," *Journal of neurophysiology*, vol. 88, no. 2, pp. 692–698, 2002.
- [42] T. Imai, K. Sekine, K. Hattori, N. Takeda, I. Koizuka, K. Nakamae, K. Miura, H. Fujioka, and T. Kubo, "Comparing the accuracy of video-oculography and the scleral search coil system in human eye movement analysis," *Auris Nasus Larynx*, vol. 32, no. 1, pp. 3–9, 2005.
- [43] T. Santini, W. Fuhl, D. Geisler, and E. Kasneci, "Eyerectoo: Open-source software for real-time pervasive head-mounted eye tracking," in *VISIGRAPP (6: VISAPP)*, pp. 96–101, 2017.
- [44] M. Pavec, J. Navratil, R. Soukup, and A. Hamacek, "A bowtie antenna prepared by aerosol jet and embroidering technology," in *2018 41st International Spring Seminar on Electronics Technology (ISSE)*, pp. 1–4, IEEE, 2018.
- [45] C. R. Picanço and F. Tonneau, "A low-cost platform for eye-tracking research: Using pupil© in behavior analysis," *Journal of the experimental analysis of behavior*, vol. 110, no. 2, pp. 157–170, 2018.
- [46] S. Anders, N. Weiskopf, D. Lule, and N. Birbaumer, "Infrared oculography—validation of a new method to monitor startle eyeblink amplitudes during fmri," *Neuroimage*, vol. 22, no. 2, pp. 767–770, 2004.
- [47] H. Kimmig, M. W. Greenlee, F. Huethe, and T. Mergner, "Mr-eyetracker: a new method for eye movement recording in functional magnetic resonance imaging," *Experimental Brain Research*, vol. 126, no. 3, pp. 443–449, 1999.
- [48] M. Khamis, F. Alt, and A. Bulling, "The past, present, and future of gaze-enabled handheld mobile devices: Survey and lessons learned," in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–17, 2018.
- [49] D. R. van Renswoude, M. E. Raijmakers, A. Koornneef, S. P. Johnson, S. Hunnius, and I. Visser, "Gazepath: An eye-tracking analysis tool that accounts for individual differences and data quality," *Behavior research methods*, vol. 50, no. 2, pp. 834–852, 2018.
- [50] S. Goni, J. Echeto, A. Villanueva, and R. Cabeza, "Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 4, pp. 941–944, IEEE, 2004.
- [51] O. V. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior research methods*, vol. 45, no. 1, pp. 203–215, 2013.
- [52] D. Sauter, B. Martin, N. Di Renzo, and C. Vomscheid, "Analysis of eye tracking movements using innovations generated by a kalman filter," *Medical and biological Engineering and Computing*, vol. 29, no. 1, pp. 63–69, 1991.
- [53] O. V. Komogortsev and J. I. Khan, "Kalman filtering in the design of eye-gaze-guided computer interfaces," in *International Conference on Human-Computer Interaction*, pp. 679–689, Springer, 2007.
- [54] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *Journal of vision*, vol. 10, no. 10, pp. 28–28, 2010.
- [55] R. Leigh and D. Zee, "The neurology of eye movements vol. 90: Oxford university press," 2015.