IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2021, accepted June 18, 2021, date of publication June 22, 2021, date of current version July 7, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3091642*

# A Human-Robot Interaction System Calculating Visual Focus of Human's Attention Level

**PARTHA CHAKRABORTY[1], (Member, IEEE), SABBIR AHMED[2], (Member, IEEE), MOHAMMAD ABU YOUSUF[2], AKM AZAD[3], SALEM A. ALYAMI[4], (Member, IEEE), AND MOHAMMAD ALI MONI[5,*]**

[1]Department of Computer Science and Engineering, Comilla University, Cumilla 3506, Bangladesh
[2]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh
[3]School of Biotechnology and Biomolecular Sciences, University of New South Wales Sydney (UNSW Sydney), Sydney, NSW 2052, Australia
[4]Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia
[5]WHO Collaborating Centre on eHealth, UNSW Digital Health, University of New South Wales Sydney (UNSW Sydney), Sydney, NSW 2052, Australia

Corresponding author: *Mohammad Ali Moni (m.moni@unsw.edu.au)

**ABSTRACT** Attention is the mental awareness of human on a particular object or a piece of information. The level of attention indicates how intense the focus is on an object or an instance. In this study, several types of human attention level have been observed. After introducing image segmentation and detection technique for facial features, eyeball movement and gaze estimation were measured. Eye movement were assessed using the video data, and a total of 10197 data instances were manually labelled for the attention level. Then Artificial Neural Network (ANN) and Recurrent Neural Network-Long Short Term Memory (LSTM) based Deep learning (DL) architectures have been proposed for analysing the data. Next, the trained DL model has been implanted into a robotic system that is capable of detecting various features; ultimately leading to the calculation of visual attention for reading, browsing, and writing purposes. This system is capable of checking the attention level of the participants and also can detect if participants are present or not. Based on a certain level of visual focus of attention (VFOA), this system interacts with the person, generates awareness and establishes verbal or visual communication with that person. The proposed ML techniques have achieved almost 99.24% validation accuracy and 99.43% test accuracy. It is also shown in the comparative study that, since the dataset volumes are limited, ANN is more suitable for attention level calculation than RNN-LSTM. We hope that the implemented robotic structure manifests the real-world implication of the proposed method.

**INDEX TERMS** Human-robot interaction, attention level, visual focus of attention, concentration, ANN, RNN-LSTM.

## I. INTRODUCTION

Human concentration differs for objects as well as time-consuming. From an academic point of view, the depth of study can be told by how deep the concentration of a student in the moments of studying. But it is wrong to think that one's mind will always be in the same way in that study. Because when someone sits down to read or pay attention to something, they become engrossed in different thoughts and the result is more or less mindfulness. As can be seen from the analysis, the main reason for the lack of mindfulness is that human thoughts are spread in different directions. This may occur due to a lot of mental pressure. Much of this article concentrated on the attention of students under different circumstances. From a student perspective, one can be under pressure for various reasons. A scenario can be depicted as a mid or final exam on the next day or a lot of big homework but it may not be possible under current circumstances. In most cases, a particular student may have to study at home or run the household. To do that, more time may be spent on tuition than studies. On the other hand, economical pressures and family earnings also create a lot of hurdles where it is difficult to meet own expenses. The responsibility of the family members, overloaded daily chores also create similar distractions. Even with great talents and skills, study concentration becomes difficult under multiple stresses and student outcomes deteriorate day by day.

On the other hand, in studies, behavioural activities often play an important role in attention, potential skill development, and many more. Recurring anxiety and depression lead

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

to lower grades and poor performance which again increases anxiety. This leads to an infinite loop and occasionally to dropout [1]. Again study environments and related elements like noise, level of comfort, number of people in one room also affect attention level. Each learning experience may aim to educate active learners but cannot meet the optimal outcomes because certain main aspects of the education design process are not taken into account. Student attributes, interior requirements elements like manner of study, age, level of maturity, and participation in developing educational environments are critical [2]. Since so many things are connected towards the attention of students an extensive survey has been done to grasp the severity and various aspects of the problem. For attention level detection eye and head movement are key features that need to be interpreted [3], Eyeball movement also plays a very important role when someone is reading, writing, or browsing an article. Whereas steady movement represents the thoughtfulness of the mind. As the movement moves from left to right or from right to leftover time, it must be understood that its attention is in order. With eyeball movement, we need to pay more attention to whether there is head movement. So, a unity system is needed where the level of attention can be detected keeping in mind the eyeball and head movement.

To define the scope and subsequent methods and estimation is required to calculate the level of VFOA for a target person when he/she is involved in reading, writing, and browsing. Initial decisions must be taken to better define important rules characteristics and care calculation based on the mathematical model. The system has been implemented with various types of people when they are reading, writing, and browsing for evaluating the level of VFOA [4]. Video clips of participants like students are taken individually with four levels of attention, high, low, average, and no attention. These video clips were further analyzed to find out the eye and head movement pattern for attention level detection. Fluctuation of attention has been tried to be analyzed in different ways. Different methods are analyzed and evaluated which was done by participants. Their attention has been categorized with a focus on the eyeball, gaze pattern, and head pose. Initial analysis has been made upon the attention level during reading, writing, and browsing. For the analysis, some scenario has been created to simulate behavioral cues during the different task. These scenario includes doing the task with full attention, background noise, body movement, phone call and habitual behavior like putting a hand on mouth, biting nail and brushing teeth with a finger, etc. [5].

The motivation of our study is to quantify the visual concentration of human attention on the current task (reading, browsing, or writing case) of the targeted participants. The major contributions of this research are:

- Collecting and Preparing a robust dataset from the visual focus of human attention based on four categories (e.g., High, Low, Average, and No Attention), which contains eyeball and gaze patterns.
- To design, fine-tune and compare several deep learning models on the created dataset and detect real-time attention using the best performing model.
- To build a robotic system to calculate the attendant's attention level that interacts with the user visually and verbally if necessary.

In this manuscript, section 2 showed the literature review of head, eyeball movement, and attention level calculation. Section 3 outlined the materials and methods, and Section 4 discussed the various analyses results and discussion. Subsequently, Section 5 presents conclusions and limitations.

## II. LITERATURE REVIEW

There are several researches that have been done on calculating the attention level and establishing human-robot interaction. Siriteerakul *et al.* [6] proposed a mathematical model for detecting head movement using image processing techniques. Mainly they used the Lucas-Kanade (LK) algorithm for pattern recognition. To recognise the position of the head, GentleBoost classifiers are used. But their main limitation was that they processed video data from low resolution-based BU datasets only, and did not check their system with noise-based video data. Martin *et al.* [7] used the movement of the head using an optical flow-based method and gesture analyzer for the detection of head movement. They named it OHMeGA analyzer. It requires the estimation of the head motion. The presented performance evaluations were under two conditions: controlled laboratory experiment and uncontrolled on-road experiment. Results show an average of 97.4% accuracy in motion states for laboratory experiments and an average of 86% accuracy overall in on-road experiments. This study lacked any motion-based video or noise-based video data analysis and did not do a fine-tuning of the model without statistical analysis. A system was proposed by the author SSA Abbas *et al.* [8] using Raspberry pi for estimating the direction of the head and track human using the Haar cascade classifier. The platform chosen for their implementation of the study was in OpenCV-Python, which is compatible with the Raspberry Pi 3. They can only count humans by head detection, but they can't track if the same person re-enters the head scene. Mane and Surve [9] have utilized computer vision methods to identify a person's head movement activity. They used KNN (K-nearest neighbor) classifier for learning. They only use the UPNA (Public University of Navarre) Head Pose Dataset for exploring head movements of many users and apply KNN and SVM with accuracy attained as 99.9% and 98.9%, respectively.

The objective of the article proposed by O'Rourke *et al.* [10], is to detect the head movement of raptors like Eagle to understand when they want to prey. Here, head movement is classified into two types: Regular head movements and Translational Head movements. They mainly focus on Cooper's Hawks, Red-tailed Hawks, and American Kestrels. However, this study didn't use any head camera

and gaze tracker. The statistical data is stored according to objects color, eye color, eye movement, the position of eyes on the head, head sideways. Various recorded video files can also be analyzed using MATLAB and OpenCV [11]. In this article various geometric and chromatic features were used as image-based features. A total number of 279 features were extracted from the video and still more features to be added in future as reported. Mittal *et al.* [12], used a distributed camera set up to monitor the driver's head movements, which detect the driver's head, provide initial estimates of the head's position, and sequentially find its position and orientation in six degrees of freedom. The primary limitation of this study is the failure of programs dependent on visual features and behavioural controls to work in low lighting conditions. Ramesh *et al.* [13] proposed an eyeball tracking and detection-based computer vision technology. The system was scripted using the MATLAB programming language. In this paper, a pupil tracking-based computer mouse movement application has been developed and implemented using a webcam. A person has to sit in front of the computer's webcam and make eyeball movements to work on the computer screen. This system has not been tested using a separate camera or a separate software system. Likewise, a triangle similarity based eyeball movement detection can be obtained when using Haar Cascade classifier for detecting the face and eyeball area using Hough transform [14]. They did not work on eye gaze tracking and did not get a success rate in the case of downward eyeball detection, where the percentage of detection success reached 79%. Prasetya *et al.* [15] proposed a method for the eye region box (both the vertical and horizontal division) using the Haar Cascade method and the KCF filter tracker with a low sensitivity level, where the computational time was low. Utaminingrum *et al.* [16] proposed a method for detecting the area of both eyes and the Hough Circle Transform of an eye gaze position. It does not work well in low-light conditions. Jyotsna *et al.* [17] proposed a stress level measurement system with Eye tracker. A stress level measurement system with Eye tracker was proposed by the author that alerts the human while detecting more blinks. It is reported that the eyeball movement works very well for stress detection. But the multi-modal stress detection system has not been applied as physiological parameters. The eye detection and tracking system was also suggested using classification techniques, and developed and tested to work with 90% double detection accuracy [18]. This system has not been tested on high-resolution-based images as different types of head motion or colored background-based images. Eye movement-based systems using HCI were used in [19], where the actual period data stream is taken via webcam using MATLAB. This system has not been tested on high-resolution-based images and only works with a webcam camera. Eye-tracking and gaze-based communication systems were presented by Kassner *et al.* [20]. It shows that Pupil can deliver a regular gaze approximation correctness of 0.6 degrees of graphic viewpoint (0.08-degree precision) with a dispensation channel potential of only 0.045 seconds. The limitation of this paper is: it works for those users who use contact lenses and eyeglasses. Bhaskar *et al.* [21] proposed how to use the independent separation associated with the optical illumination flow. The independent fragmentation allows for the rapid determination of possible travel locations. They accomplished a 97.0% achievement rate in sensing blink using the proposed method. This system takes a lot of time for blink-detection, which is a major obstacle to eyeball detection. Alam *et al.* [22] proposed a Euclidean distance-based thresholding and geometric slope-based technique for gesture recognition with an overall accuracy of 91.95%. In addition to RGB cameras, 3D camera modules can be used for recognition. The same author also detected trajectories [23] with 99.32% accuracy using LSTM and Convolutional Neural Network (CNN). In the detection and segmentation paradigm, image-based surveillance systems play a significant role. Mondal *et al.* [24] examined several tracking and classification-based surveillance systems as well as their benefits and drawbacks. The authors mentioned challenges such as unusual orientation, presence in occlusion, and variation in illumination, all of which are important for recognizing human attention. Human attention can be extracted from the gesture and position of the human body, particularly from the eyes. Dutta *et al.* [25] has provided a comprehensive review on vision tracking algorithms. To achieve a certain level of accuracy, machine learning methods required appropriate parameters such as loss function optimizers. According to a recent study [26], Adam is one of the most useful optimizers. Again, LSTM is used for classification and analysis tasks due to its unique ability to remember and forget parts of data [27].

A visual attention-based system was proposed by Yang *et al.* [28] using Eye movement indicators, where ANOVA as well as t-test investigation were employed. For multimedia presentations, attention is calculated based on eyeball movements, where the presence of images and text on a slide increases the attention and the presence of images alone reduces the attention. The limitation in this study is that it has been done only with the available data of the students through a multimedia slide presentation. Voice examination and the head-turning system were introduced by the author in the paper [29], where the author used a mixture of when speaking among persons, where the detection of head-turning sector's success rate was 87.7%. The ROMEO2 project has been implemented to create a human-robot interaction system whose job is to work as an assistant to the people with loss of autonomy. The main limitation of this work is that their dataset was small while the audio and video cues weren't combined. Comparatively, Frutos-Pascual *et al.* [30] calculated the attention skill from various children aged between 8 to 12 years using their gaze shapes and communication with eye-tracking instruments. The author achieved 88% classification accuracy using a random forest classifier. The main limitation is that they work with very little data (only 32 children in Spain), and the processing speed of the system is slower. Another paper has also used almost

similar approaches of eye-tracking technology in the case of calculating the students' visual attention from a web-based interface [31]. Among 500 students, 50 out of 10 groups were tested (based on Force Concept Inventory or FCI question on Physics) on the web system and attention level calculation was done through eye tracker. This system is based on problem-solving purposes only. Peng *et al.* [32] suggested the Interest Meter (IM), a system that adopts blink discovery, jerk detection, head gesture detection, and facial appearance appreciation to measure users' attention. The primary limitation of this study is that different types of head orientation are not recognized here. In the paper [33], thirty-five images that contain numerous parts of interest were shown to ten applicants, where the eye movements were documented using a head-mounted EyeLink-II system, the key issue here is how the construction site workers can keep their minds on hazardous situations. They have worked with only 12 workers and a half to collect data by analyzing in various ways. Viola-Jones object detection algorithm can also be utilized for similar facial feature detection [34]. The algorithm is trained with some attentive and inattentive faces, and had 14% false positives when the alt-tree training file was used. In the context of this study and the proposed practical use of the technology, this is considered low enough. In the case of detecting faces, images with or without faces are required, and the dataset needed to be large for accurate prediction. A real-time monitoring system [35] was proposed to detect driver fatigue/drowsiness, where the Haar Cascade file is used for detecting the face. The main limitation of this study is that the system fails to work if there is a lack of light and the driver wears sunglasses. The AdaBoost algorithm can adjust the weight of the sample according to the classification error rate of the weak classifier. The AdaBoost algorithm constructs a solid classifier with the lowest error rate. The human eye detection algorithm based on AdaBoost is suggested in another paper [36]. Boosting is a fusion of multiple classifiers that progress the classification presentation of the common methods. The essence of the boosting algorithm is to use a different subset for iterative training on weak classifiers and conduct weighted training on samples until higher accuracy is achieved. To indicate the level of attentiveness, the percentage age of eye closure (PERCLOS) system was introduced by the author Dasgupta *et al.* [37]. They also used Haar-like features for face recognition and the Kalman Filter for monitoring. The system proposed by Canedo [38] uses computer vision methods for observing classrooms. The system can perceive the face detection correctness rate with low resolution in a classroom setting. The main limitation of this study is that if there are any participants outside the fixed position sitting on the chair, their attention cannot be calculated. Moreover, a static database has been used in that study for student identification. Mastronardi *et al.* [39] proposed a system that facilitates distance learning, which means it provides education beyond the physical barriers. It uniquely identifies the users. It implements facial detection and recognition. The main limitation of this study is that this system works by creating a database of everyone in advance with a static bio-metric approach. No hybrid approach has been adopted, which is a major limitation in that study. Another study [40] proposes driver drowsiness and distraction-based systems while the driver is looking ahead. The main limitation of that study is that fewer users were tested in this system, where the test time was much shorter. The author proposed multiple choice-based eye-tracking systems [41], while calculating visual attention using MATLAB programming language. Again, an eye-tracking based student's visual attention-based system is implemented by Klein *et al.* [42], where the student's mean response sureness result for right answers (74%) was higher compared to incorrect answers (67%). The main limitation of both of these studies is that due to the multiple choice-based question and answer system, everyone has to pay more attention while answering, and a lot of time has been wasted to select the answer, which slows down the performance of the system. Facial appearance and emotions-based structure was proposed using an API formed by OpenCV [43]. The main limitation of this study is that the system is built with a simple webcam and GUI-based C# desktop application, which does not work on mobile systems and does not work with sunglasses. Masse *et al.* [44] explain how to track the gaze and use visual focus of attention (VFOA) based approach in the context of social contact. The author proposes a Bayesian state-space model that takes advantage of the association between head motions and eye gaze, and on the other hand, determines the correlation between the visual concentration of attention and eye gaze. The biggest drawback here is that when two persons are looking at each other, the attention level cannot be computed. In another article, Hung *et al.* [32] proposed an Interest meter system to measure users' reactions and interest. Head and eye movement and related features have been the main focus of this article Context-aware algorithm has been utilized to generate interest scores and user emotional state.

In our earlier study [4], we have established a human–robot interaction system to detect the visual focus of attention (VFOA) based on human attention (in case of both reading and browsing purposes). The system detects the person's current task (attention) and estimates the level by detecting the head and estimating the eye region, especially detected iris center within the eye area (gaze pattern calculation). The system also determines the interest or willingness of the target person to interact based on a certain level of VFOA. Then, depending on the level of interest of the target person, the system generates awareness and establishes a communication channel with him/her.

Moreover, our other study [5] has offered various machine learning approaches for predicting the level of visual focus of a human's attention. Using data collected from survey reports (environmental data) and eyeball data (eyeball movement, reading time, and head turn) while reading an article, a dataset with each participant's attention level was formed. Eight different classifiers Logistic Regression, Support

Vector Machine (SVM), Decision Tree, K-Nearest Neighbor (KNN), AdaBoost, Multilayer Perceptron (MLP), Extra Tree classifier, and Voting Classifier were trained for classifying the participant's attention level into three classes: High, Average, and Low. In that study [5], the Logistic Regression achieved the highest accuracy of 96% outperforming others in predicting these classes, whereas the aggregated weighted Voting Classifier attained an accuracy of 95%. As a reported future work of that study [5], here in this work, we are interested to employ deep neural networks since it has the potential to further enhance the system's performance to near-human level judgements.

## III. MATERIALS AND METHODS

Here we present our proposed deep-learning based framework for human-robot interaction system as depicted in Fig. 1. The overall methodology is composed of several components. First, from the robotic interfaces and sensors, necessary real-time image frames were captured. Next, the eyeball and head have been extracted from the image frames, which was followed by a gaze estimation through the eyeball movement. If the participant is constantly seeing something, then it was understood that he had no attention. Otherwise, the attention level is calculated and if the system finds that attention level as 'low' during checking, then it initiates a verbal communication with the participant. On the other hand, the system remains noninteractive with the participant.

### A. DATA COLLECTION

To measure the impact of VFOA, a group of student participants (where the range of a participant's age is 20 to 31, and the average age is 24) has been video-recorded while studying (i.e. reading and writing) and web browsing very intensely at different times of a day. From 400 participants, a complete dataset of video recordings was constructed with a total of 2705, 2668, and 3165 for three categories, i.g. reading, writing, and browsing, respectively. The data in each of those categories were then further classified as 'Low', 'Average', 'High', and 'No', based on how long a participant's attention had been on their work. The number of eye movements, head movements, and time elapsed in seconds for each of those tasks had been calculated [see the Detection and Segmentation sections below] and collected in a spreadsheet after labelling them according to the attention level. In total, the dataset contains **10197** values; where **2954** values of no attention, **1961** values of low attention, **1659** values of average attention and **3623** values of high attention has been labelled. The labels for the attention level 'none', 'low', 'average', and 'high' were categorically *one-hot* encoded for four output. Table 1 represents the data distribution among different attention levels and activity. For training any classifier algorithm, the data was split into two-part, training and testing sets. The testing set contained 20% randomly selected data, and the training set contained the remaining 80%.

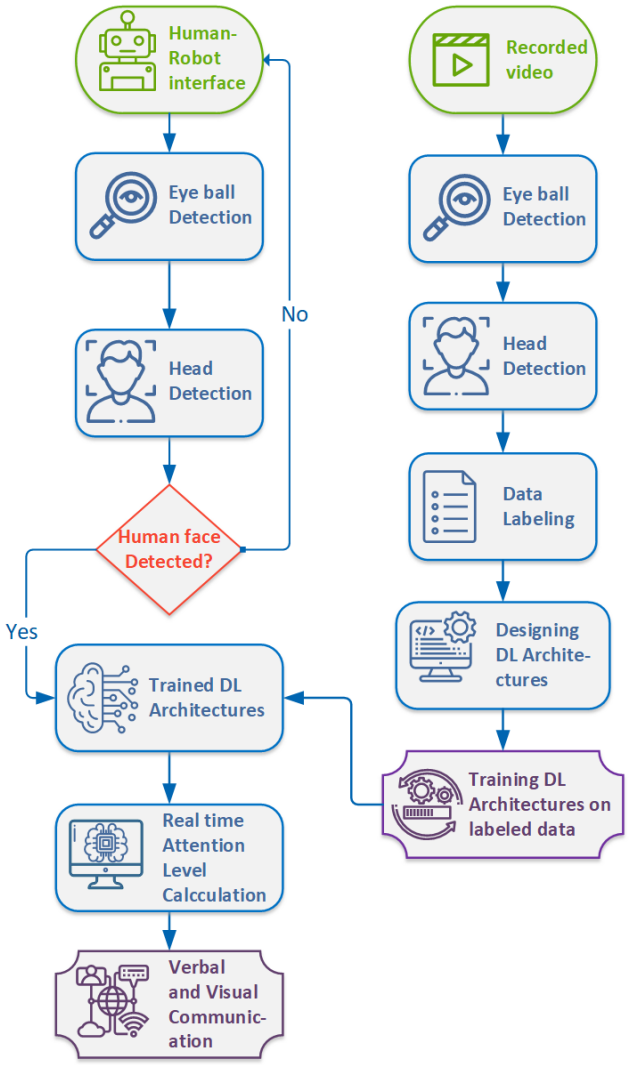



**FIGURE 1.** Schematic diagram of our proposed human-robot interaction system empowered by deep learning models.

**TABLE 1.** Classification of collected video recordings of participants' reading, writing and browsing based on their attention level.

| Category | No | Low | Average | High | **Total** |
|---|---|---|---|---|---|
| Reading (400) | 997 | 632 | 670 | 1076 | 3375 |
| Writing (400) | 864 | 602 | 476 | 1202 | 3144 |
| Browsing (400) | 1093 | 727 | 513 | 1345 | 3678 |
| **Total** | 2954 | 1961 | 1659 | 3623 | **10197** |

### B. ARCHITECTURAL DEVELOPMENT

#### 1) HUMAN-ROBOT INTERACTION SYSEM

The human-robot interaction system was built with a modular robotic infrastructure that is equipped with local computation and power systems. For computational units, a Raspberry pi 3B+ has been used. The computational unit is consists of several separate blocks that co-operated with each other. These hardware blocks are for video capture, camera movement, real-time output through the LCD panel, and a sound system for verbal communication. A physical structure was built with an aluminium frame and additional material to accommodate

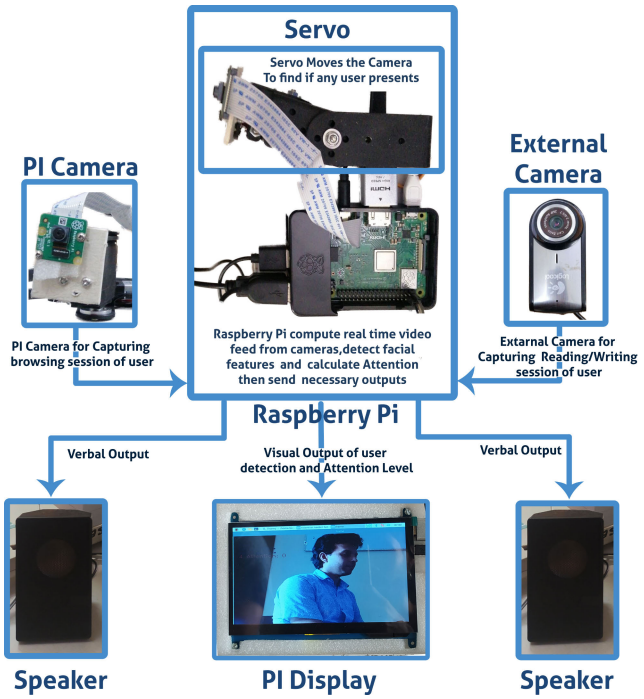



**FIGURE 2.** Various components and their inter-relation in robotic interface.

necessary parts. An additional AC power supply also was added for the wall plug. This was also supported by a DC (battery) power supply in the case of modular operation. Fig. 2 depicts the overall diagram of the robotic interface, for which the detailed description are presented below:

### 2) VIDEO CAPTURE MECHANISM

The first challenge for capturing a participant's video was that the distance between robotic cameras the user should be optimal, as attention level may not be correctly predicted if its facial area is not visible enough due to distance from the camera. To resolve this issue, two camera modules were used as the video capture mechanism. Using a proprietary connection, a Raspberry Pi camera is wired directly to the Raspberry Pi. Another USB webcam is connected via a USB-A port on the Raspberry Pi. The Pi camera has been strictly used for the web browsing attention level calculation whereas a USB camera has been used for reading and writing attention level calculation since it can easily be placed with the book, reading, and writing material. A demonstration of this video-capture module is shown in Fig 3.

#### *a: CAMERA MOVEMENT PROCEDURE*

The Pi Camera was set on the top of the servo motor, which allows for horizontal scanning. This scanning consists of a 10 Degree step size i.e: the servo moves 10 degrees at a time.

Then the camera captured frames and extracted facial features within the images based on the discussed methods in the following section. A threshold value has been set to filter how many frames to be considered to the algorithm before any movement. If user $= 0$, that indicates that there is no

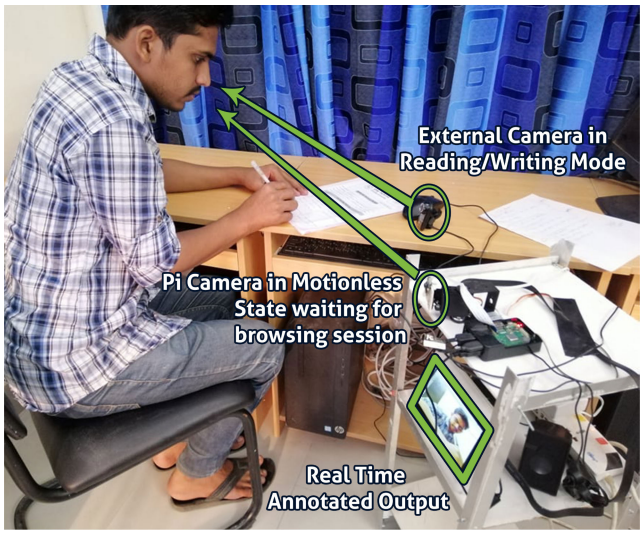



**FIGURE 3.** Video capture mechanism for reading, writing and browsing session using Pi camera and external USB camera.

**Algorithm 1** An Algorithm for Camera Movement

```
Angle ← 0
Increment ← 10
Thrs ← 5
while true do
  Count ← 0
  Capture_Frame ← image_frame_from_video
  Count ← Count + 1
  User ← CountUser(Capture_frame)
  if User == 0 then
    if Count ≥ Thrs then
      Angle = Angle + Increment
      RotateServo( Angle )
      if Angle < 10 then
        Increment = 10
      else if Angle > 170 then
        Increment = −10
      end if
    else
      go to CaptureFrame()
    end if
  end if
end while
```

user in the current frame. Then we move the camera angle slightly using servo motors and capture frames again to find out if any user is present or not. This stage is repeated until the user is found in the captured frame. Algorithm 1 depicts the overall procedure for camera movement and scans a user for attention level calculation. The USB camera has been stationary and could be attached to the target participant's reading or writing materials. If there's no user present at the current frames, the servo will move again 10 degrees in the same direction. The servo has been set to scan for 0 degrees to 180 degrees, and then back to 0 degrees. Raspberry pi

camera has a field of view of 62 degrees. 10 degrees has been selected as the movement angle. The device has been implemented keeping the mindset of human recognition up to 3 meters and the human head is approximately 17 cm in width. Now 'T' is the arc length of a circle of s = r*x, where the radius is r and the central angle is x in radians, so, the Degree = 0.17*170/3 = 10.2. Hence, the rotation for each turn is selected as 10 degrees.

#### 3) VISUAL OUTPUT

Visual Output is achieved with a seven-inch Raspberry Pi display. This display uses an HDMI connection for receiving signals from the Raspberry Pi. Hence an HDMI to HDMI cable has been employed to connect both ends. The video feed captured from the cameras is directly streamed to the display with the annotated eyeball, head movement, where the user looking at, and their current attention level.

#### 4) VERBAL COMMUNICATION

Since the robotic system has insufficient computing power, a voice message mechanism is introduced for added convenience, instead of text-to-speech. This component can only play pre-recorded messages, which are saved in the system's internal memory. In the case of a user being out of the camera frame, a voice message has been set to alert the user. Again if the user's attention level becomes too low, another message for increased concentration has been set to be played.

### C. DETECTION AND SEGMENTATION

#### 1) PUPIL POSITION DETECTION

Detection and segmentation of facial features are compulsory for the precise calculation of eyeball and chin movement. The histogram of oriented gradients (HOG) is an ideal feature of the facial image [45]. HOG-based feature detection is implemented in the 'DLIB' library, which is used for head and eye movement. Since it is widely used for object detection, and in the robotic system, frames were captured when the camera was in a stationary position. In factors influenced by an object, the technique counts instances of gradient orientation. The gradient has been calculated with orientation binning and turned into descriptor blocks. Then these blocks were normalised and used for training a support vector machine classifier was used to be trained. The 'Dlib' [4], a python library is used for this purpose, which is capable of calculating HOG-based features. In this work, the shape68 predictor [4] has been implemented to match the bounding box coordinates from the detection algorithm. Shape68 is a pre-trained landmark detector of the Dlib library. It calculates 68 coordinate points to map facial features. Dlib library has been used to isolate the face and eye from the rest of the image. Then the image was blurred to smooth and reduce the noise if exists. Next, it was converted into black and white followed by the contour estimation of the pupil.

The black and white conversion requires a threshold value, where, if any pixel is greater than this threshold value, then the pixel will be set as white, otherwise the pixel will be set as black. Since each image can be captured with different lighting conditions and scenarios, this threshold value also needs to be automated. Thus an adaptive algorithm is devised to calculate the threshold value where the mean pixel value of the first few frames is calculated and that ought to be the threshold value. Fig. 4 represent above-mentioned method for pupil coordinate calculation.

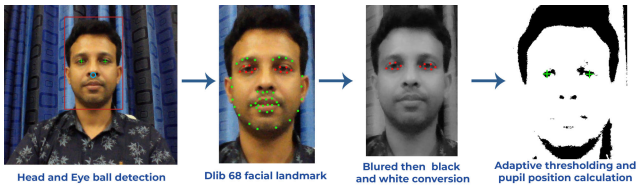



**FIGURE 4.** Pupil position detection using face detection, eye segmentation and thresholding.

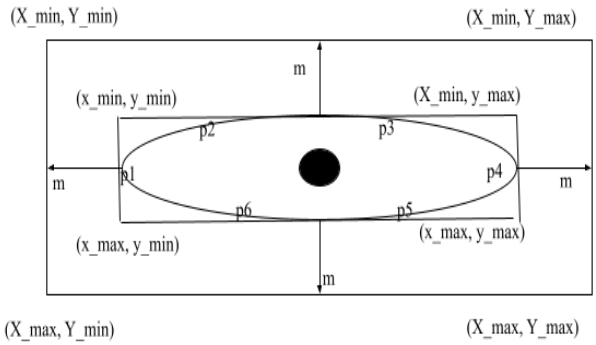



**FIGURE 5.** Extraction of eye using 6 point.

#### 2) EYE MOVEMENT

The overall eye movement calculation part is divided into three parts: eye extraction, pupil's centroid coordinate detection, and eyeball movement counting.

##### *a: EXTRACT EYE*

Using shape68 predictor [4], we get 6 border coordinates P = p1, p2, p3, p4, p5, p6 of each eye in Fig. 5. We calculated the minimum and maximum values for the x-axis points and y-axis points from these coordinates.

$$x_{min} = min(x\_axis\_points\_of\_P) \quad (1)$$

$$x_{max} = max(x\_axis\_points\_of\_P) \quad (2)$$

$$y_{min} = min(y\_axis\_points\_of\_P) \quad (3)$$

$$y_{max} = max(y\_axis\_points\_of\_P) \quad (4)$$

Note, $(x_{min}, y_{min})$, $(x_{min}, y_{max})$, $(x_{max}, y_{min})$ and $(x_{max}, y_{max})$ are the corner of the smallest rectangle, which can cover the full eye [Fig. 5]. Next, we added a margin, m = 5 around each of the four sides.

$$X_{min} = x_{min} - m \quad (5)$$

$$X_{max} = X_{max} + m \tag{6}$$
$$Y_{min} = y_{min} - m \tag{7}$$
$$Y_{max} = Y_{max} + m \tag{8}$$

Finally, we achieved the coordinates $(x_{min}, y_{min})$, $(x_{min}, y_{max})$, $(x_{max}, y_{min})$ and $(x_{max}, y_{max})$ to extract a full eye and create a new frame with the features of that isolated eye. Hence, the coordinates of the the eye center was considered to be the center of the new frame.

$$(x_{center}, y_{center}) = (height/2, width/2), \tag{9}$$

where height and width are the dimensions of the eye frame and $(x_{center}, y_{center})$ is the coordinate of the center of the eye frame.

#### b: DETECTING PUPIL'S CENTROID COORDINATE

After getting the isolated eye, we conducted several steps to detect the coordinate of the iris. At first, we slightly blurred the image to remove any kind of noise. To remove the backlight, we erode the image. After that, we binarized the image to get only black and white pixels. Here, we considered the black pixels as the contour. Finally, we calculated the centroid of the contour and considered that value as the position of the pupil's centroid coordinate.

#### c: COUNT EYE MOVEMENT

To count the eye movement, we check whether the participant was looking left or right. If it was looking at the left side on the current frame but was looking at the right side on the previous frame, we considered that incident as eye movement. To find out this, we calculated the horizontal ratio of the pupils of both eyes, which can be calculated by the following formula:

$$hr = \frac{x_{pupil}}{2 * x_{center} - 10}, \tag{10}$$

where *hr* is the horizontal ratio. If the average of the horizontal ratio of two eyes was greater than 65%, then we consider that the person is looking to the right, and if the average horizontal ratio is less than 35%, then it is looking to the left. The situation between 35% and 65% indicates that the participant is looking at the center.

### 3) HEAD MOVEMENT

At first, we detect four coordinates (p1, p2, p3, p4) from a frame using shape68 detector in the Fig. 6. Where p1 is the midpoint of the two eyes, p2 is the leftmost point of the left cheek, p3 is the rightmost point of the right cheek and p4 is the bottom-most coordinate of the chin. Next, we calculated the pair-wise distances for (p1, p2), (p2, p4), (p1, p3), and (p3, p4) by the following formula:

$$dis = \sqrt{(x - x1)^2) + (y - y1)^2} \tag{11}$$

If the summation of distance from p2 to p1 and p4 is greater than the summation of distance from p3 to p1 and p4, then the participant is looking at the left part of the screen, and he was looking at the right part of the screen otherwise. Same as 'eye movement count', If the participant is looking at the left side on the current frame and also at the right side on the previous frame, we consider this incident as a head movement.

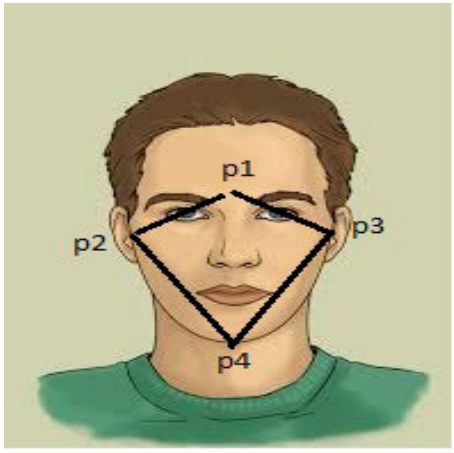



**FIGURE 6.** Head movement detection using 4 facial landmark point.

## D. MODEL DEVELOPMENT

### 1) ANN

Here, in this study, an Artificial Neural Network (ANN) was used for modelling attention level estimation. Different layers were stacked in a sequential style to predict the output. The particular ANN model has the following layers:

**Dense Layer:** Dense layers are fully connected to the previous layers. Each neuron of a dense layer is connected to every neuron in the following layer. The value of each neuron is multiplied by some weighted numbers, and then summarized in subsequent neurons. This weighted value changes according to a given input and output data. An activation function, typically Rectified Linear Unit (ReLU), was applied in the hidden layers. *Softmax* or *sigmoid* activation function typically applied to the output dense layer for prediction, where in this study we have used the *softmax* function.

**Batch normalization:** The summation of weighted value can vary significantly after a finite amount of data being trained. Thus it creates highly expensive computation as well as overflow. To overcome this issue, *re-centering* and *rescaling* were applied to the input layers. *Batch normalization* undertakes similar procedures with a batch size to stabilize the learning process.

**Dropout layer:** This is a regularisation technique to reduce or remove oscillating weights. Drastic changes in weight values may indicate that the network is unstable or the corresponding neurons have a lower effect on the detection procedure. Thus the dropout layer was applied to avoid over-fitting problems.

The proposed ANN architecture consists of a total of 8 layers including *dense* and *batch normalization layers*. The dropout layer has been deducted in the fine-tuning process. Since the three input attributes (i.e., 'Time duration',

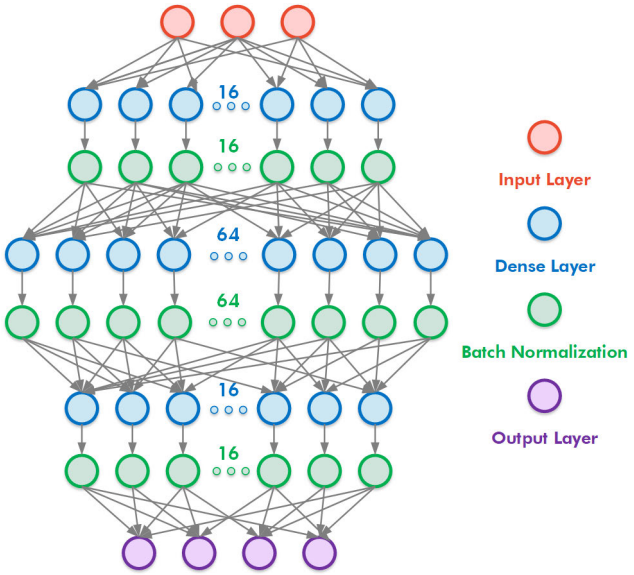



**FIGURE 7.** Proposed ANN architecture layer organisation and parameter.

'Head movement', and 'Eye Movement') and four output labels (i.e., 'Low', 'Average', 'High', and 'None') have been defined in our problem statement, the input dense layer contains three neurons. Three hidden dense layers with batch normalization have been utilised subsequently. Each of the dense layers contains 16, 64, 16 neurons with ReLU activation. Intermediate batch normalization increases the stability and performance of the network. The output dense layer utilizes the *softmax* as an activation function. *Adam* has been used in architecture as an optimiser. The proposed ANN architectures description is given in Fig. 7

### 2) RNN-LSTM

In addition the ANN, a *Long Short Term Memory (LSTM)*-based Recurrent Neural Network (RNN) also has been implemented for attention level detection. RNN is typically used in time series data but it can also be used in classification tasks [5]. As they take data from previous inputs to affect the present input and output, they are characterized by their memories. A variant of an RNN is a long short-term memory network (LSTM) network that helps to efficiently record past knowledge in memories. The network's sensitivity to inputs decreases over time when new inputs replace the activation of the hidden layer and the network tends to forget the first inputs, a phenomenon known as the 'vanishing gradient' problem. An LSTM network is similar to a regular RNN, although the hidden layer summation units are substituted by memory blocks. The multiplicative gates enable LSTM memory cells to preserve and obtain information over extended periods, reducing the vanishing gradient. In one LSTM cell, three gates were present: an *input* gate with *tanh* and *sigmoid* activation, a *forget* gate with *sigmoid* activation, and an *output* gate with *sigmoid* and *tanh* activation. Proposed RNN-LSTM networks contain several layers. The *input* layer has three units since the input dataset contains three columns.

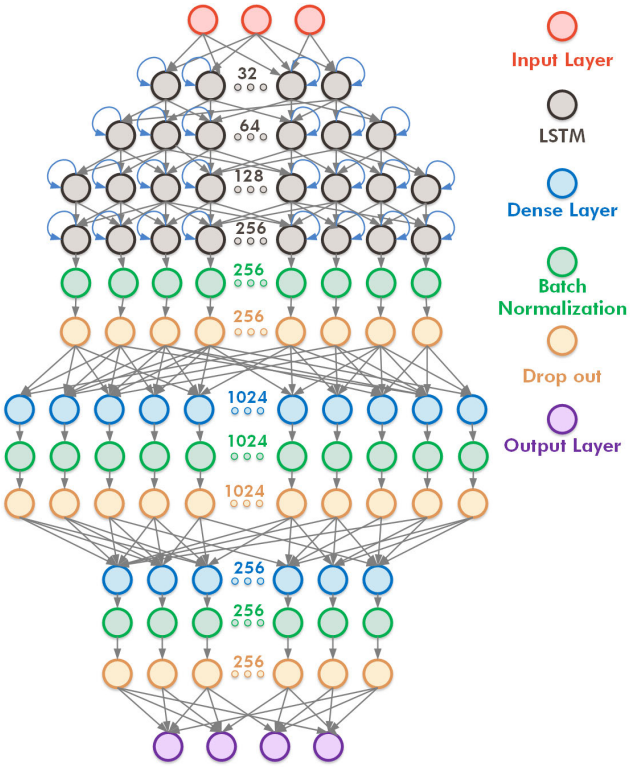



**FIGURE 8.** Proposed RNN-LSTM architecture layer organisation and parameter.

Subsequent four LSTM layers were been added to the model with 32, 64, 128, 256 units respectively. 10% drop out was also added to each LSTM layer to reduce over-fitting. These LSTM layers have been set to return sequence to the previous layer if necessary.

Next, to conduct classification, *fully connected* layers were added subsequently. Rectified linear units (ReLu) was used as and activation functions in dense layers. *Dropout* and *batch normalization* also added to reduce over-fitting issues. The sequences of *fully connected, drop out* and *batch normalization* repeated two times, then another dense layer is added as output with four units. Sigmoid was used as an *output* layer activation function. The 'Adam optimiser' was used as an optimiser function along with the *binary cross-entropy* as the *loss* function for the proposed architecture. The proposed RNN-LSTM architectures description is depicted in Fig. 8.

## IV. EXPERIMENTAL RESULT AND DISCUSSION

The experiment has been conducted using two devices. The initial training was conducted on a separate computer equipped with an intel core i7 processor, 16 GB ram, and 4GB Nvidia 920mx GPU. Both ANN and RNN architectures have been trained with the trial and error basis for fine-tuning the model. A *mini-batch* of size 500 was used to run the calculation on a low-specification computer. The final trained and tested model was loaded into the Raspberry Pi 3 module for attention calculation. Raspberry pi cameras and separate webcams were used for capturing video. An additional servo

motor has been added for manoeuvring and a sound system for playing voice message.

### A. EVALUATION MEASUREMENTS

To evaluate the performance, at first we calculated the confusion matrix. The confusion matrix compares predicted output to actual output and calculates true positive (TP), true negative (TN), false positive (FP) and false negative (FN), and based on those values, the several performance metrics are evaluated, which are described below:

**Accuracy** Accuracy is the measurement of how many predictions are true including both true positive and true negative. Which is then divided by the total number of predictions. Accuracy represents the percentage of both correctly predicted instances that match with the true output class. Accuracy values depend on not only the class that should be correct but also not predicting other classes as correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

**MSLE** Mean squared logarithmic error (MSLE) is the average of the squared difference between logarithmic predicted (pr) and logarithmic actual (ac) output. MSLE penalizes models by estimating logarithmic differences, and can also be interpreted as a correct prediction ratio. For taking logarithmic error both small and larger difference in prediction and true class, MSLE penalizes both types of error approximately the same.

$$MSLE = \frac{\sum(log(pr) - log(ac))^2}{n} \tag{13}$$

**MAE** Mean absolute error is simply the absolute value of the difference between predicted (pr) and actual (ac) output. MAE is useful if both of the compared data instances have the same scaling, in this article which is predicted and actual attention level.

$$MAE = frac\sum |pr - ac|n \tag{14}$$

**Recall** Recall represents the percentage of positive label retrieval with total predicted and actual positive output. In the scope of attention level, among all the user, it represents how many of those are truly attentive by the prediction.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

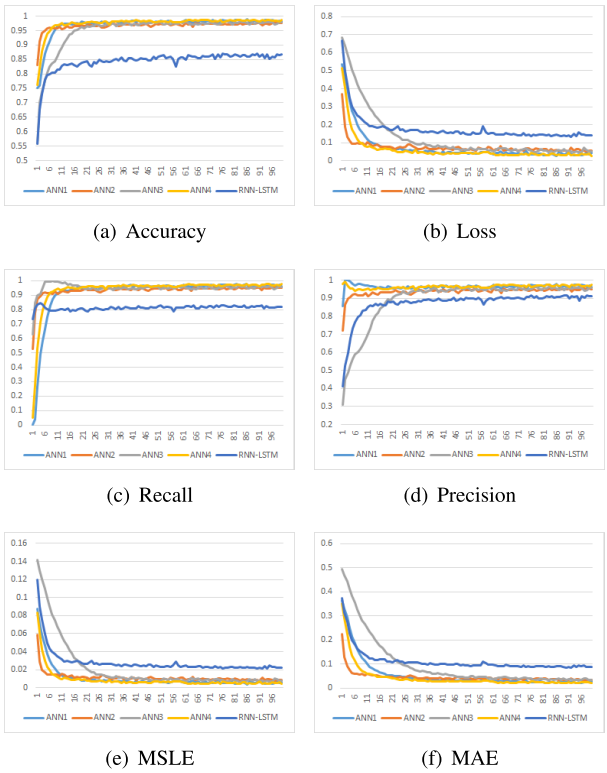
**Precision** Precision represents the percentage of positive correct predicted output. It is the ratio between predicted true and total prediction. The question precision answer in this scope is: among the predicted user who has a certain level of attention, how many are true.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

**F1-Score** F1-Score is the harmonic average of precision and recall where Score adds extra weights, accurate evaluation of precision or recall rather than the other. For attention level detection, predictor performance mostly depends on the correctly predicted true class. Since false negative and false positive both are taken into account in F1-Score, it is rather more useful than other statistical measures.

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{17}$$

**AUC** The AUC of a model describes how well it can discriminate between classes. AUC represents aggregate measurements of the true positive and true negative rates. True positive prediction is weighted more heavily than true negative prediction by AUC.




(a) Accuracy (b) Loss

(c) Recall (d) Precision

(e) MSLE (f) MAE

**FIGURE 9.** Epoch-wise evaluation measurements of ANN, ANN2, ANN3, ANN4 and LSTM.

### B. COMPARATIVE ANALYSIS

Since customized data is used for the experiment, direct performance comparison with other recent literature is unavailable. For this reason, several ANN and RNN models have been designed to find out the final model with the best performance. Each of the models has been trained for 100 epochs. In these experiments, the ANN models perform better than the RNN-LSTM model. Thus multiple ANN models and their results have been calculated to find out the optimised model. Different models that contribute to the selection of the better model are given descriptions of the parameters and statistical measurements Fig. 9 respectively are below:

**ANN1:** Total three hidden dense layers with 16, 64, 16 units, and three *batch normalization* were used in this model. The output dense layer had *softmax* activation, the *loss* function was the *binary cross-entropy* and the optimiser

was Adam. After training, it achieved a validation accuracy of 99.24%, F1-Score of 0.984, and AUC of 0.999.

**ANN2:** Total four hidden dense layers with 16, 256, 512, 256 units, and four *batch normalization* and dropout were used in this model. The amount of dropout was 25%. The output dense layer had *softmax* activation, the *loss* function was *binary cross-entropy* and the optimiser was Adam. After training, it achieved validation accuracy of 98.38%, F1-Score of 0.968, and AUC of 0.999.

**ANN3:** Total three hidden dense layers with 16, 64, 16 units, and three *batch normalization* were used in this model. Output dense layer had *sigmoid* activation, the *loss* function was the *binary cross-entropy* and the optimiser was *rmsprop*. After training, it achieved a validation accuracy of 98.37%, F1-Score of 0.967, and AUC of 0.999.

**ANN4:** Total three hidden dense layers with 16, 128, 32 units, and three *batch normalization* were used in this model. The output dense layer had *softmax* activation, the *loss* function was *categorical cross-entropy* and the optimiser was Adam. After training it achieved a validation accuracy of 99.15%, F1-Score of 0.983, and AUC of 0.999.

**RNN-LSTM:** Total four LSTM layers and two dense layers, *batch normalization*, and dropout were used in this model. The output dense layer had *sigmoid* activation, the *loss* function was *binary cross-entropy* and the optimiser was the Adam. After training, it achieved a validation accuracy of 97.79%, F1-Score of 0.961, and AUC of 0.991.

### C. DISCUSSION

The deep learning model has been designed upon the initial data analysis discussed in the previous section. Since input contains three variable and output consists of four classes, to compute their correlation, hidden layers have been utilized between the input and output. Weights on nodes and edges allow for the adjustment of communication signal strengths, that can be strengthened or reduced by a prolonged period. ANNs should project the test data based on the training and subsequent adaptation of the matrices, node, and edge weights. Batch normalization has been added with a hidden layer to cope with overflow and over-fitting. RNN-LSTM model has been also designed to find out possible relations with a time dimension and input data. Since LSTM layers are computationally expensive, dropouts are added for reducing unnecessary neurons. A trial and error method has been followed to set the different parameters, neurons, and layer numbers. These parameters include different loss functions such as binary cross-entropy, categorical cross-entropy, mean squared logarithmic error, mean absolute error; and different optimiser algorithms such as rmsprop and Adam. After fine-tuning, the best performing model with its performances metrics has been found.

#### 1) TIME COMPLEXITY

Let $N_1$, $N_2$, $N_3$, $N_4$ $\ldots N_n$ are the neuron numbers in each layer. Then the complexity of any neural network can be

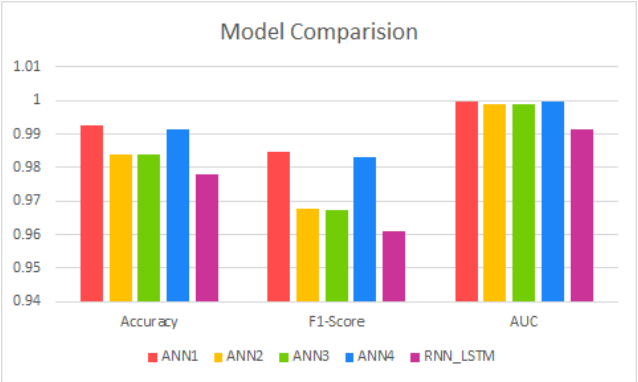



**FIGURE 10.** Performance comparison of the models.

defined as:

$$O(N) = E * T * (N_1 * N_2 + N_2 * N_3 + \cdots + N_{n-1} * N_n) \tag{18}$$

where, Epoch (E) = 100 and Train Size (T) = 10197

$$\begin{aligned}
&\text{ANN1} : 100 * 10197 * (3 * 16 + 16 * 64 \\
&\quad + 64 * 16 + 16 * 4) = 2.2 * 10^9 \\
&\text{ANN2} : 100 * 10197 * (3 * 16 + 16 * 256 + 256 * 512 \\
&\quad + 512 * 256 + 256 * 4) = 2.72 * 10^{11} \\
&\text{ANN3} : 100 * 10197 * (3 * 16 + 16 * 64 \\
&\quad + 64 * 16 + 16 * 4) = 2.2 * 10^9 \\
&\text{ANN4} : 100 * 10197 * (3 * 16 + 16 * 128 \\
&\quad + 128 * 32 + 16 * 4) = 6.44 * 10^9 \\
&\text{LSTM} : 100 * 10197 * (3 * 32 + 32 * 64 + 64 * 128 \\
&\quad + 128 * 256 + 256 * 1024 + 1024 * 256 \\
&\quad + 256 * 3) = 5.79 * 10^{11}
\end{aligned}$$

From the complexity perspective, the LSTM model has the highest complexity although it has lower test and validation accuracy than other models. The ANN1 model has been calculated as the lowest time complex model among all other models. Though LSTM provides more robustness due to its unique characteristic to process time-series data, it suffers lower statistical performance due to the smaller dataset, input, and output size. If the inclusion of time series data can be omitted, ANN1 training metrics closely follow the validation metrics, which probe to more robustness. The difference between ANN1 and ANN3 is that ANN1 has been implemented with 'Adam' [26] and ANN3 has been implemented with 'rmsprop' optimizers, where ANN1 superior results reflect the former one to be the best suited as an optimizer in this context. The results can be further improved by using more epochs and further fine-tuning.

To compare various ANN and LSTM models, the validation accuracy, F1-Score, and AUC of different models are estimated and depicted in Fig. 10. Training accuracy is omitted because a different number of parameters makes

models prone to under-fit, which in turn creates higher training results. It can be easily observed from the figure that ANN1 performance is better than other models. LSTM-RNN models have performed poorly since the dataset only contains time-independent structured data.

The deep learning model has been saved with the corresponding weights to implement in a real-world scenario. As described in methodology the ANN saved model then implanted into the raspberry pi device to conduct head detection, pupil detection, and movement algorithms. The human-robot interaction system has been deployed to specify the applicability of the proposed machine learning models and overall robotic features. The rest of the mechanism in the system mostly rule-based features that enable easier transition and communication between a user and the systems. Though it can be prone to slower response than the regular computer as a result of using raspberry pi with multiple modules and algorithms. Two directional scanning using servo motors enables searching and locking of user position, thus the system can be Independent of being positioned in certain places. Users can better understand the system's output by using verbal and visual communication.

## V. CONCLUSION

In this study, we have analyzed the reading, writing and browsing categories of videos from a range of participants and created a dataset. A deep learning-based architecture has been proposed, which is used to categories the dataset into four classes of attention level (low, average, high, and none). A robotic system has been implemented with the trained model to calculate attention level in real-life scenario. Since the performances of deep learning algorithms depend on a large amount of data, we postulate that a dataset larger than our's could lead to better prediction. Another limitation of the proposed system is that it used various libraries for detection and segmentation. Completely custom algorithm design may also provide better results, which are to be solved in our future works. Nonetheless, this research work provides multiple aspects of deep learning and human-machine interfaces for human attention level calculation for further integration into human-robot interaction.

## REFERENCES

[1] D. Eisenberg, M. F. Downs, E. Golberstein, and K. Zivin, "Stigma and help seeking for mental health among college students," *Med. Care Res. Rev.*, vol. 66, no. 5, pp. 522–541, Oct. 2009.

[2] M. Yilmaz-Soylu and B. Akkoyunlu, "The effect of learning styles on achievement in different learning environments," *Turkish Online J. Educ. Technol.*, vol. 8, no. 4, pp. 43–50, 2009.

[3] J. M. Henderson, "Visual attention and eye movement control during reading and picture viewing," in *Eye Movements Vision Cognition*. New York, NY, USA: Springer, 1992, pp. 260–283.

[4] P. Chakraborty, M. A. Yousuf, and N. Faruqui, "How can a robot calculate the level of visual focus of human's attention," in *Proc. Int. Joint Conf. Comput. Intell.* Singapore: Springer, 2020, pp. 329–342.

[5] P. Chakraborty, M. A. Yousuf, and S. Rahman, "Predicting level of visual focus of human's attention using machine learning approaches," in *Proc. Int. Conf. Trends Comput. Cognit. Eng.*, M. S. Kaiser, A. Bandyopadhyay, M. Mahmud, and K. Ray, Eds. Singapore: Springer, 2021, pp. 683–694.

[6] T. Siriteerakul, Y. Sato, and V. Boonjing, "Estimating change in head pose from low resolution video using LBP-based tracking," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2011, pp. 1–6.

[7] S. Martin, C. Tran, A. Tawari, J. Kwan, and M. Trivedi, "Optical flow based head movement and gesture analysis in automotive environment," in *Proc. 15th Int. Conf. Intell. Transp. Syst.*, 2012, pp. 882–887.

[8] S. S. A. Abbas, M. Anitha, and X. V. Jaini, "Realization of multiple human head detection and direction movement using raspberry pi," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 1160–1164.

[9] S. S. Mane and A. R. Surve, "Engagement detection using video-based estimation of head movement," in *Proc. 3rd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2018, pp. 1745–1749.

[10] C. T. O'Rourke, T. Pitlik, M. Hoover, and E. Fernández-Juricic, "Hawk eyes II: Diurnal raptors differ in head movement strategies when scanning from perches," *PLoS ONE*, vol. 5, no. 9, Sep. 2010, Art. no. e12169.

[11] H. Monkaresi, M. S. Hussain, and R. A. Calvo, "Classification of affects using head movement, skin color features and physiological signals," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2012, pp. 2664–2669.

[12] A. Mittal, K. Kumar, S. Dhamija, and M. Kaur, "Head movement-based driver drowsiness detection: A review of state-of-art techniques," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, Mar. 2016, pp. 903–908.

[13] R. R. Rishikesh M, "Eye ball movement to control computer screen," *J. Biosens. Bioelectron.*, vol. 6, no. 3, p. 1, 2015.

[14] R. P. Prasetya and F. Utaminingrum, "Triangle similarity approach for detecting eyeball movement," in *Proc. 5th Int. Symp. Comput. Bus. Intell. (ISCBI)*, Aug. 2017, pp. 37–40.

[15] R. Prasetya, F. Utaminingrum, and W. Mahmudy, "Real time eyeball movement detection based on region division and midpoint position," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 3, pp. 149–158, Jun. 2018.

[16] F. Utaminingrum, M. A. Fauzi, Y. A. Sari, R. Primaswara, and S. Adinugroho, "Eye movement as navigator for disabled person," in *Proc. Int. Conf. Commun. Inf. Syst. (ICCIS)*, 2016, pp. 1–5.

[17] C. Jyotsna and J. Amudha, "Eye gaze as an indicator for stress level analysis in students," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, 2018, pp. 1588–1593.

[18] Z. Al-Kassim and Q. A. Memon, "Designing a low-cost eyeball tracking keyboard for paralyzed people," *Comput. Electr. Eng.*, vol. 58, pp. 20–29, Feb. 2017.

[19] O. Mazhar, T. A. Shah, M. A. Khan, and S. Tehami, "A real-time Webcam based eye ball tracking system using MATLAB," in *Proc. IEEE 21st Int. Symp. Design Technol. Electron. Packag. (SIITME)*, Oct. 2015, pp. 139–142.

[20] M. Kassner, W. Patera, and A. Bulling, "Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput., Adjunct*, Sep. 2014, pp. 1151–1160.

[21] T. N. Bhaskar, F. Tun Keat, S. Ranganath, and Y. V. Venkatesh, "Blink detection and eye tracking for eye localization," in *Proc. Conf. Convergent Technol. Asia–Pacific Region*, 2003, pp. 821–824.

[22] M. S. Alam, K.-C. Kwon, and N. Kim, "Implementation of a character recognition system based on finger-joint tracking using a depth camera," *IEEE Trans. Human-Mach. Syst.*, vol. 51, no. 3, pp. 229–241, Jun. 2021.

[23] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, p. 376, Jan. 2020.

[24] A. Mondal, A. Dutta, N. Dey, and S. Sen, "Visual traffic surveillance: A concise survey," in *Information Technology and Intelligent Transportation Systems*. Amsterdam, The Netherlands: IOS Press, 2020, pp. 32–41.

[25] A. Dutta, A. Mondal, N. Dey, S. Sen, L. Moraru, and A. E. Hassanien, "Vision tracking: A survey of the state-of-the-art," *Social Netw. Comput. Sci.*, vol. 1, no. 1, pp. 1–19, Jan. 2020.

[26] M. N. Y. Ali, M. G. Sarowar, M. L. Rahman, J. Chaki, N. Dey, and J. M. R. S. Tavares, "Adam deep learning with SOM for human sentiment classification," *Int. J. Ambient Comput. Intell.*, vol. 10, no. 3, pp. 92–116, Jul. 2019.

[27] S. Sivakumar and R. Rajalakshmi, "Analysis of sentiment on movie reviews using word embedding self-attentive LSTM," *Int. J. Ambient Comput. Intell.*, vol. 12, no. 2, pp. 33–52, Apr. 2021.

[28] F.-Y. Yang, C.-Y. Chang, W.-R. Chien, Y.-T. Chien, and Y.-H. Tseng, "Tracking learners' visual attention during a multimedia presentation in a real classroom," *Comput. Educ.*, vol. 62, pp. 208–220, Mar. 2013.

[29] M. E. A. Sehili, F. Yang, and L. Devillers, "Attention detection in elderly people-robot spoken interaction," in *Proc. Workshop Multimodal, Multi-Party, Real-World Hum.-Robot Interact.*, 2014, pp. 7–12.

[30] M. Frutos-Pascual and B. Garcia-Zapirain, "Assessing visual attention using eye tracking sensors in intelligent cognitive therapies based on serious games," *Sensors*, vol. 15, no. 5, pp. 11092–11117, May 2015.
[31] J. Han, L. Chen, Z. Fu, J. Fritchman, and L. Bao, "Eye-tracking of visual attention in Web-based assessment using the force concept inventory," *Eur. J. Phys.*, vol. 38, no. 4, Jul. 2017, Art. no. 045702.
[32] W.-T. Peng, C.-H. Chang, W.-T. Chu, W.-J. Huang, C.-N. Chou, and W.-Y. Chang, "A real-time user interest meter and its applications in home video summarizing," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2010, pp. 849–854.
[33] S. A. Bhoir, S. Hasanzadeh, B. Esmaeili, M. D. Dodd, and M. S. Fardhosseini, "Measuring construction workers' attention using eye-tracking technology," in *Proc. 11th Construct. Specialty Conf.*, Vancouver, BC, Canada, 2015, pp. 222-1–222-10.
[34] J. Cöster and M. Ohlsson, "Human attention: The possibility of measuring human attention using OpenCV and the Viola-Jones face detection algorithm," Dept. Comput. Sci. Eng., Kth Roy. Inst. Technol. CSC, Degree Project, First Level, Stockholm, Sweden, Tech. Rep., 2015.
[35] J. Suryaprasad, D. Sandesh, V. Saraswathi, D. Swathi, and S. Manjunath, "Real time drowsy driver detection using haarcascade samples," in *Proc. Comput. Sci. Inf. Technol.*, 2013, pp. 45–54.
[36] L. Shang, C. Zhang, and G. Gao, "Eye detection and attention recognition based on OPENCV," *DEStech Trans. Comput. Sci. Eng.*, vol. 1, pp. 465–468, Jun. 2017.
[37] A. Dasgupta, A. George, S. L. Happy, and A. Routray, "A vision-based system for monitoring the loss of attention in automotive drivers," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1825–1838, Dec. 2013.
[38] D. Canedo, A. Trifan, and A. J. Neves, "Monitoring students' attention in a classroom through computer vision," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst.* Wiesbaden, Germany: Springer, 2018, pp. 371–378.
[39] G. Mastronardi, V. Bevilacqua, R. F. Depasquale, and M. D. F. Vilardi, "Attention control during distance learning sessions," in *Proc. Int. Conf. Image Anal. Process.* Berlin, Germany: Springer, 2013, pp. 545–549.
[40] J. Jo, "Vision-based method for detecting driver drowsiness and distraction in driver monitoring system," *Opt. Eng.*, vol. 50, no. 12, Dec. 2011, Art. no. 127202.
[41] M.-J. Tsai, H.-T. Hou, M.-L. Lai, W.-Y. Liu, and F.-Y. Yang, "Visual attention for solving multiple-choice science problem: An eye-tracking analysis," *Comput. Educ.*, vol. 58, no. 1, pp. 375–385, Jan. 2012.
[42] P. Klein, A. Lichtenberger, S. Kächemann, S. Becker, M. Kekule, J. Viiri, C. Baadte, A. Vaterlaus, and J. Kuhn, "Visual attention while solving the test of understanding graphs in kinematics: An eye-tracking analysis," *Eur. J. Phys.*, vol. 41, no. 2, Mar. 2020, Art. no. 025701.
[43] A. Valdestilhas, F. I. Masseto, and A. Ferreira, "An adaptive graphical user interface based on attention level and diameter of eye pupil," in *Proc. IEEE 6th Int. Conf. Adapt. Sci. Technol. (ICAST)*, Oct. 2014, pp. 1–4.
[44] B. Masse, S. Ba, and R. Horaud, "Tracking gaze and visual focus of attention of people involved in social interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2711–2724, Nov. 2018.
[45] Y. Pang, Y. Yuan, X. Li, and J. Pan, "Efficient HOG human detection," *Signal Process.*, vol. 91, no. 4, pp. 773–781, Apr. 2011.

**PARTHA CHAKRABORTY** (Member, IEEE) received the bachelor's and master's degrees in computer science and engineering from Jahangirnagar University, Dhaka, Bangladesh. He has seven years of experience in teaching as well as in research. He is currently working as the Head of Department and an Assistant Professor with the Department of Computer Science and Engineering (CSE), Comilla University, Cumilla, Bangladesh. He has authored more than 40 research articles from various national and international conferences and journals to date. He has delivered invited talks in various national and international webinar sessions. His research interests include human–robot interaction, computer vision, image processing, machine learning, and NLP. He has associated with some conferences throughout Bangladesh as a TPC member and a reviewer of various national and international conferences and journals. As a Professional IEEE Member, he participates in the Professional Bodies/Societies' research activities. For his contributions to several research initiatives, he has given the Comilla University research fund, as well as a special fund from the Bangladesh Government's ICT Ministry.

**SABBIR AHMED** (Member, IEEE) is currently pursuing the bachelor's degree with the Institute of Information Technology, Jahangirnagar University. He is also fulfilling his duty as a Teaching Assistant and a Research Assistant with the Institute of Information Technology. He is also completing his responsibility as the Vice-Chair of the IEEE Student Branch, Jahangirnagar University. He is also a Competitive Programmer and also a Robotics Enthusiast. His research interests include image processing, robotics, the IoT, machine learning, and algorithms.

**MOHAMMAD ABU YOUSUF** received the B.Sc. (engineering) degree in computer science and engineering from the Shahjalal University of Science and Technology, Sylhet, Bangladesh, in 1999, the M.E. degree in biomedical engineering from Kyung Hee University, South Korea, in 2009, and the Ph.D. degree in science and engineering from Saitama University, Japan, in 2013. In 2003, he joined the Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh, as a Lecturer. In 2014, he moved to the Institute of Information Technology, Jahangirnagar University. He is currently working as a Professor with the Institute of Information Technology, Jahangirnagar University, Dhaka, Bangladesh. His research interests include medical image processing, human–robot interaction, computer vision, and natural language processing.

**AKM AZAD** received the Ph.D. degree in computational systems biology and biostatistics from Monash University, Australia, in 2017, followed by his first postdoctoral fellowship in the Faculty of Information Technology, Monash University. He was a Postdoctoral Research Associate with the AI Lab, School of Biotechnology and Biomolecular Sciences (BABS), UNSW Sydney, contributing/leading several bioinformatics and computational biology projects. Recently, he joined the University of Technology Sydney as a Research Fellow for developing cutting-edge methodologies and tools for online inference of phylogenetic trees from COVID-19 sequences. His research interests include AI/ML/DL, Bayesian statistics, MCMC, Bayesian networks, bioinformatics, and computational biology.

**SALEM A. ALYAMI** (Member, IEEE) received the Ph.D. degree in biostatistics from Monash University, Australia, in 2017. He has been an Assistance Professor with the School of Mathematics and Statistics, IMAMU, Riyadh, Saudi Arabia, since 2017, contributing/leading several grants in biostatistics projects. Recently, he has appointed as the Dean of the Deanship of Scientific Research with IMAMU. His research interests include Bayesian networks, neural networks, Bayesian statistics, MCMC methods, and applications of statistics in biology and medicine.

**MOHAMMAD ALI MONI** received the Ph.D. degree in artificial intelligence and machine learning from the University of Cambridge, U.K., in 2014. From 2015 to 2017, he was a Postdoctoral Research Fellow with the Garvan Institute of Medical Research, Sydney, NSW, Australia. He was also an Associate Lecturer with the University of New South Wales Sydney (UNSW Sydney), Australia. At the end of 2017, he was awarded the USyd Fellowship with The University of Sydney. He is currently working as a Research Fellow and a Conjoint-Lecturer with UNSW Sydney. His research interests include artificial intelligence, machine learning, data science, and clinical bioinformatics.