Received July 17, 2020, accepted July 25, 2020, date of publication July 31, 2020, date of current version August 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013540

Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey

ANDRONICUS A. AKINYELU AND PIETER BLIGNAUT
Department of Computer Science and Informatics, University of the Free State, Bloemfontein 9301, South Africa
Corresponding author: Andronicus A. Akinyelu (akinyeluaa@ufs.ac.za)

This work was supported by the University of the Free State, South Africa.

ABSTRACT Eye tracking is becoming a very important tool across many domains, including human-computer-interaction, psychology, computer vision, and medical diagnosis. Different methods have been used to tackle eye tracking, however, some of them are inaccurate under real-world conditions, while some require explicit user calibration which can be burdensome. Some of these methods suffer from poor image quality and variable light conditions. The recent success and prevalence of deep learning have greatly improved the performance of eye-tracking. The availability of large-scale datasets has further improved the performance of deep learning-based methods. This article presents a survey of the current state-of-the-art on deep learning-based gaze estimation techniques, with a focus on Convolutional Neural Networks (CNN). This article also provides a survey on other machine learning-based gaze estimation techniques. This study aims to empower the research community with valuable and useful insights that can enhance the design and development of improved and efficient deep learning-based eye-tracking models. This study also provides information on various pre-trained models, network architectures, and open-source datasets that are useful for training deep learning models.

INDEX TERMS Convolutional neural network, deep learning, eye tracking, eye movements, gaze estimation, computer vision, region of interest.

I. INTRODUCTION

Currently, different types of commercial and non-commercial eye-tracking solutions exist, including model-based and appearance-based methods; however, some of these solutions are expensive or inaccurate under real-world conditions, while some require explicit user calibration which can be burdensome [1]. Hence, recent eye-tracking studies are focusing on developing deep learning-based eye-tracking techniques that do not require explicit user calibration [1]. With the recent rise of deep learning, Convolutional Neural Network (CNN) based gaze estimation models are becoming very popular and prevalent. It became popular when LeCun, *et al.* [2] successfully applied them to hand-written digit classification. CNN models are also suitable for handling large scale datasets, and they have been successfully applied to many different domains, such as computer vision [3], speech recognition [4], and language modelling [5]. CNN models are capable of directly mapping image features, such as pupil and glint locations, to gaze points without the use of hand-engineered features [6].

This article presents a survey of appearance-based gaze estimation techniques, with a focus on CNNs. This article also presents a survey of other machine learning-based gaze estimation techniques. The primary highlights of this study are as follows:

a. This article presents a survey of recent CNN-based gaze estimation techniques to provide researchers with valuable and useful insights that can enhance the design of improved and efficient deep learning-based eye-tracking models.
b. This article provides a discussion on the performance of various gaze estimation models. It also provides useful insights on various large-scale datasets, network architectures, and pre-trained models that are useful for building improved deep learning-based gaze estimation models.

This article will empower the research community with a gaze estimation kit that can enhance the design of improved gaze estimation techniques. This article could serve as a reference and starting point for researchers seeking to design efficient deep learning-based techniques.

II. BASIC CONCEPTS

This section provides an overview of some fundamental concepts that are related to this study; specifically gaze estimation and CNN.

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

A. GAZE ESTIMATION

Gaze estimation is a technology that aims to understand the intent and interest of users [7]. Gaze estimation techniques focus on the relationship between the image data and gaze direction [8]. Gaze directions are estimated based on specific eye features (such as pupil and corneal reflection) extracted from the eye regions of image data collected from single or multiple cameras [9]. Generally, gaze estimation techniques can be grouped into two: model-based and appearance-based techniques [8]. More information on the two techniques is provided in the next sub-sections. Some practical applications of gaze estimation include mobile gaze interactions [10], [11], driver gaze monitoring [3], [12], on-screen keyboard typing [13], and Virtual Reality (VR) [14].

1) APPEARANCE-BASED GAZE ESTIMATION

Appearance-based techniques rely on the photometric appearance of the eye to perform gaze estimation [9]. They generally need a single camera to capture eye images which are then used to build gaze estimation models that can map the appearances of the captured images to certain gaze directions. These models do not require hand-engineered features for training; they can extract image features implicitly from the data. Compared to model-based techniques, appearance-based techniques require a larger number of eye images for training, and this gives them the capacity to learn invariance in appearance disparity [15].

Appearance-based gaze estimation models are considered to produce good results in real-world settings [11]. They are designed to directly predict screen coordinates, implying that they can only be used for a single device and orientation [16], [17]. This is because their gaze location is directly defined in the coordinate system of the target screen [18]. Some studies designed techniques that can predict gaze location relative to the camera [10], [19]. The gaze location of these techniques is defined as a virtual plane in the camera coordinate system [18].

2) MODEL-BASED GAZE ESTIMATION

Model-based techniques perform gaze estimation by combining the geometric model of the eye with eye features, such as cornea reflection and pupil centre [20]. Figure 1 shows the geometric model of the eye. As shown in the figure, the optical axis is defined as the line that connects the cornea centre to the pupil centre. The gaze direction is determined by the visual axis, which goes through the cornea centre and the fovea. The point of regard (PoR) can be defined as the intersection between the visual axis and the display surface [1]. The angle kappa is defined as the angle between the visual axis and the optical axis. The angle difference between the optical and visual axis is expressed as $\theta = (\alpha, \beta)$, where θ is a constant vector for each individual [1]. Personal calibration for model-based techniques is typically used to estimate the value of θ for each subject.

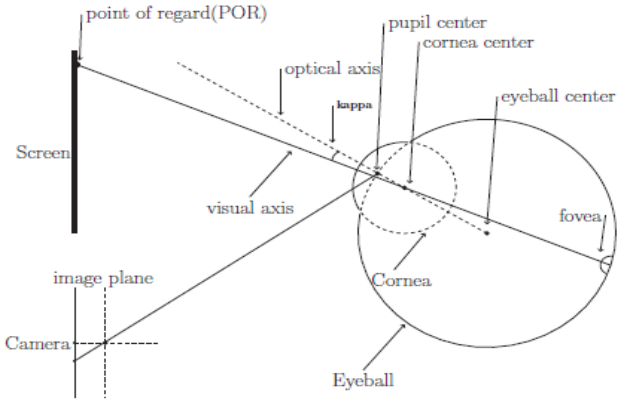


FIGURE 1. 3D eye model [1].

3) PERFORMANCE METRICS FOR GAZE ESTIMATION

In the literature, gaze tracking accuracy metrics have been reported in different ways, including angular accuracy in degrees [1], [14], [21], distance accuracy in cm/mm [10], [22], and gaze estimation accuracy in percentage [3], [13], [23].

In this article, the terms accuracy and prediction error are used to describe the performance of various gaze estimation systems. Accuracy is mostly reported for classification problems, such as driver gaze zone classification and eye movement classification. It is used to measure the percentage of correctly classified categories, based on the number of frames in the evaluation dataset. As an example, for driver gaze zone classification, accuracy measures how well the gaze zones of each frame (in the evaluation dataset) are correctly classified. A correctly classified frame refers to the frame where its predicted gaze zone is equal to the actual gaze zone. Prediction error is mostly reported for regression problems. It is used to measure how well a model predicts the gaze coordinates. It shows the difference between the predicted and actual gaze coordinates. Some calculations for gaze estimation accuracy are given below [24]:

a: GAZE POINT COORDINATES IN PIXELS

The calculation for the gaze point coordinates in pixels can be obtained from (1) and (2)

$$Gaze\_X = mean\left(\frac{x_{left} + x_{right}}{2}\right) \tag{1}$$

$$Gaze\_Y = mean\left(\frac{Y_{left} + Y_{right}}{2}\right) \tag{2}$$

where $(X_{left}, Y_{left}, X_{right}, Y_{right})$ represents the actual gaze coordinates of the left and right eye as obtained from the eye tracker.

b: GAZE POSITION IN mm

The gaze position in mm of on-screen distance can be calculated by using (3) to (4):

$$X\_Position(mm) = \mu * Gaze\_X \tag{3}$$

$$Y\_Position(mm) = \mu * Gaze\_Y \tag{4}$$

where μ is the pixel size of the monitor that is used for eye tracking. It is calculated based on the dimension and pixel resolution of the monitor's screen. The calculation for μ is can be obtained from (5)

$$\mu = dim_m / dim_p \tag{5}$$

where dim_m is the diagonal size of the screen measured in mm (originally in inches), and dim_p is the diagonal size of the screen measured in pixels as shown in (6).

$$dim_p = \sqrt{width_p^2 + height_p^2} \tag{6}$$

where $width_p$ *and* $height_p$ is the width and height of the screen respectively, measured in pixel.

c: ON-SCREEN DISTANCE

If the origin of the gaze coordinate system is located at (X_{pixels}, Y_{pixels}), then as shown in (7), at the bottom of the next page, the on-screen distance of the gaze point of a user is the distance between the origin and a specific gaze point. The offset is defined as the distance between the sensor of the eye tracker and the lower edge of the display screen. If the tracker is attached directly below the screen, then the offset is 0 and the origin is the centre of the screen. That is, $X_{pixels}, y_{pixels} = (0, 0)$.

d: GAZE ANGLE RELATIVE TO THE EYE

The gaze angle of a point on the screen relative to the eye of a user can be obtained using (8).

$$gaze\ angle(\theta) = \tan^{-1} OSDist/Z \tag{8}$$

where Z is the distance of the eye from the screen. The distance between the eye and the gaze point on the screen is estimated as shown in (9):

$$EstGP(mm) = \sqrt{\left((Gaze\_X)^2 + (Gaze\_Y)^2 + (Z)^2\right)} \tag{9}$$

e: PIXEL DISTANCE

During calibration, some points are displayed on the screen and users are asked to look at the displayed points. The (x, y) coordinates displayed on the screen form the ground truth for the captured images. The shift between the ground truth coordinates (GT_x, GT_y) and the estimated gaze points $(Gaze\_X, Gaze\_Y)$ can be calculated using (10).

$$pix\_shift(pixels) = \sqrt{\left((GT_x - Gaze\_X)^2 + \left(GT_y - Gaze\_Y\right)^2\right)} \tag{10}$$

f: ANGULAR ACCURACY

The gaze estimation accuracy (or prediction error) of an eye tracker (in degrees) can be expressed as the angular deviation between the actual and the estimated gaze locations. Using equations (8) to (10), the prediction error can be calculated using (11).

$$ang\_acc = \left(\mu * pix\_shift * \cos(mean(\theta))^2\right) / EstGP \tag{11}$$

g: EUCLIDEAN DISTANCE IN cm/mm

Some studies [10], [19] used the Euclidean metric to evaluate the accuracy of their techniques. They reported the Average Euclidean Distance (AED) from the location of true fixation. The average Euclidean distance between the estimated gaze coordinates (x, y) and the ground truth gaze coordinates can be obtained from (12):

$$\text{AED} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(gt\_x_i - e\_x_i)^2 + (gt\_y_i - e\_y_i)^2} \tag{12}$$

where gt_x_i *and* e_y_i refers to the ground truth label for each input, and e_x_i *and* e_y_i refers to the estimated (x, y) gaze coordinates for each input.

h: STRICTLY CORRECT ESTIMATION RATE

Gaze accuracy can be measured based on Strictly Correct Estimation Rate (SCER) [3]. SCER can be obtained by dividing the number of strictly correct frames by the total number of frames. The strictly correct frames refer to the frames where the predicted gaze zone is equal to the actual gaze zone.

i: LOOSELY CORRECT ESTIMATION RATE

Gaze accuracy can be measured based on loosely correct estimation rate (LCER) [3]. LCER can be obtained by dividing the number of loosely correct frames by the total number of frames. The loosely correct frames refer to the frames where the predicted gaze zone is around the surrounding region of the actual gaze zone.

More information on gaze estimation performance metrics can be found in [24]. Most of the performance metrics used in the literature are not similar and cannot be compared to each other, hence it is the decision of developers to choose a suitable performance measure. Generally, there is no standard performance measure used for benchmarking the performance of deep learning models. This makes the comparison between gaze estimation systems difficult. Lemley *et al.* [22] argues that angular resolution is the most consistent because it properly describes the performance of an algorithm regardless of other system parameters, such as screen pixel size or distance from eye tracker. Kar and Corcoran [20] proposed a framework that can be used to practically evaluate different gaze estimation techniques.

B. CONVOLUTIONAL NEURAL NETWORKS

In deep learning, CNNs are a class of deep neural networks that are mostly applied to evaluate visual images [25]. They are inspired by the organization of the visual cortex in the brain [22]. The visual cortex is the main region of the brain that receives and processes visual information transmitted from the eye [26]. As shown in Figure 2, CNNs are very similar to regular neural networks made up of different neurons with learnable weights and biases. Each neuron in the

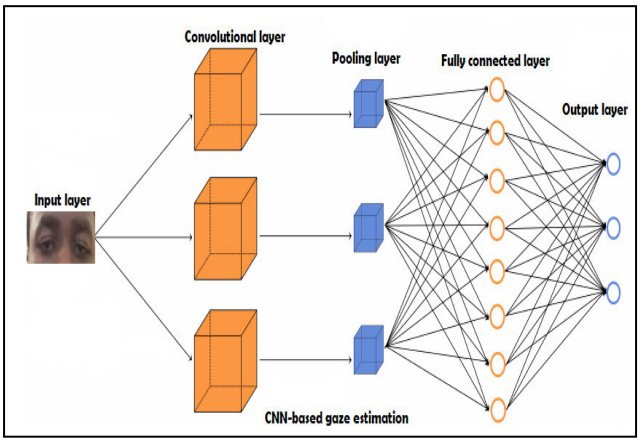


FIGURE 2. A regular 3-layer neural network and a regular convolutional neural network [28].

network accept an input, performs a dot product operation, and optionally follow the dot operation with a non-linearity operation (such as Rectified Linear Unit (ReLU)). CNN architectures make the explicit assumption that all inputs are images and thus allow users to encode certain properties into the architecture [27]. Typically, a CNN consists of multiple convolutional layers followed by pooling, non-linearity, and finally fully connected layer(s) plus output layer. The first three layers are responsible for feature extraction, while the fully connected layer is responsible for classification [13].

1) CONOLUTIONAL LAYER

The convolutional layer consists of a set of trainable parameters that in turn consist of a set of learnable filters. Each filter (also called kernel) is small in width and height, but they extend through the entire depth of the input volume [27]. When an input is forwarded into a network, each filter is convolved across the width and height of the input volume, and the dot operation is computed between the entries of the filter and the entries of the input using (13) [29]. As the filters are slid over the width and height of the input volume, a 2-dimensional activation map (or feature map) is produced.

$$H[i,j]=(m*k)[i,j]=\sum_a\sum_b k[a,b]m[i-a,j-b] \tag{13}$$

where m represents the input image and k represent the kernel, a and b represent the row and column of the resultant matrix.

2) POOLING LAYER

The pooling layer performs a down-sampling operation along the width and height of the input, resulting in a reduction in the volume dimension. There are different types of pooling techniques, including max-pooling and average pooling. Both operations are performed by applying a filter to a non-overlapping subregion of an input. For example, as shown in Figure 1, if our input is a 4×4 matrix, and we are sliding a 2×2 filter over the input, we will have a stride of 2. For each region represented by the 2×2 filter, we will take the maximum (for max-pooling) or average (for average-pooling) of that region and create a new matrix, where each entry in the new matrix is the maximum or average of the 2×2 regions in the 4×4 matrix. As shown in Figure 3, the result for sliding the filter over the input without overlapping any region of the input will be a 2×2 matrix.

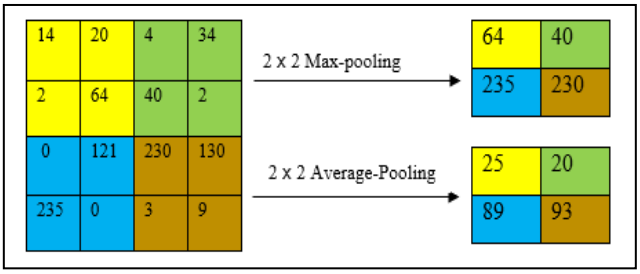


FIGURE 3. Max pooling and average pooling example.

3) NON-LINEARITY LAYER

The non-linearity layer is responsible for applying an elementwise activation function on the input. An activation function is used to determine whether a neuron in the network should be activated or not. The non-linear activation performs a non-linear transformation of the input making it possible for a network to learn complex relationships in datasets. There are different types of non-linear activation function, including sigmoid, tanh, Rectified Linear Unit (ReLu), and softmax. The ReLu activation function is the most used because it produces good results. The ReLu activation function is defined mathematically in equation (14) and shown in Figure 4. As shown, the ReLu function is linear for all positive values and non-linear for negative values; it outputs all negative values as zero.

$$relu(x) = max(0,x) \tag{14}$$

where x is the input of the function

4) FULLY CONNECTED LAYER

The fully connected layer performs the final classification or regression. It is composed of different neurons, like the neurons in artificial neural networks. The output of one layer of neurons is computed using (15) [30]. A fully connected layer takes the output (i.e. feature map matrix) from the previous

$$OSDist(mm) = \mu\sqrt{\left(\left(Gaze_x - \frac{x_{pixels}}{2}\right)^2 + \left(y_{pixels} - Gaze\_Y + \frac{offset}{pixelsize}\right)^2\right)} \tag{7}$$

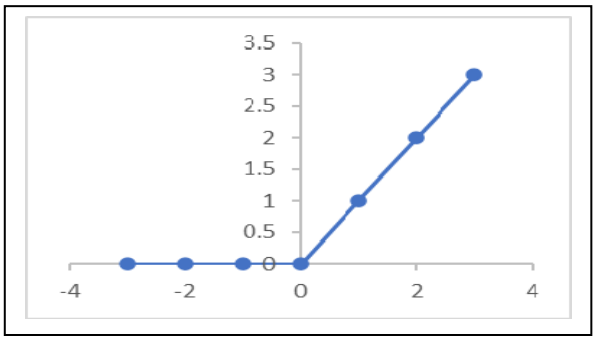


FIGURE 4. ReLu activation function.

layers and converts them into a single vector (or 1D array) that can be used as input to the layer. The 1D array is then passed through one or more fully connected layer(s). The fully connected layer performs linear transformation and non-linear transformation on the 1D array. The linear transformation is performed using (15), while the non-linear transformation is performed using equation (16).

$$Z = W^T \cdot X + b \tag{15}$$

where X refers to the input, b refers to the bias, and W refers to the learnable parameters (or weights) in the network. W is a matrix of randomly initialized numbers.

$$output = \sigma(Z) \tag{16}$$

where σ refers to the activation function (such as ReLU), and Z refers to the output of the linear transformation operation.

5) OUTPUT LAYER

The last fully-connected layer (or output layer) is finally passed through an activation function to produce the final output, which could be a continuous value if we are solving a regression problem or categorical value if we are solving a classification problem.

In this fashion, CNN transforms an input image layer by layer from the initial pixel values to the final classification or regression. The convolutional layers and fully connected layers are the trainable layers, and their transformations are based on their various trainable weights, biases, and activations. The non-linearity and pooling layers are not trainable as they simply perform fixed functions. The parameters in the convolutional and fully connected layers are trained using a gradient descent algorithm so that the final output of the network will correspond to the labels that are used to train the network. More details on the operations of CNN can be seen in [31]. A CNN can be applied to inputs of any dimension, however, due to their success in 2D images, they are mostly applied to 2D inputs plus the colour channels [22].

III. SURVEY OF GAZE ESTIMATION METHODS

Different appearance-based gaze estimation studies have been designed for 2D [11], [18], 3D gaze estimation [18], [32], [33], classification and regression. Earlier studies focused on Artificial Neural Networks (ANNs) [34], [35], random forest [21], [33], linear regression [36], support vector regression (SVRs) [19], multimodal models [11], [32], deep end-to-end CNN models [3], [10], incremental learning [37] and transfer learning [38]. Some studies introduced hybrid appearance-based techniques. For example, Zhang, *et al.* [11], Wang, *et al.* [21] combined classical feature extraction techniques with ANN regressors. Different network architectures were designed in the literature. Some networks were designed to accept only eye images [8], [11], [21], [36], while other networks were designed to accept both eyes and head pose feature vector to simultaneously model free head movement [10], [11]. Recent studies combined full-face images with CNN models for gaze estimation [3], [39].

The task of designing a deep learning model for gaze estimation can be considered as a regression or classification task. Although both are very useful, regression provides the best prediction flexibility [22] thanks to its ability to find gaze angles that correspond to eye images. This section presents a survey of appearance-based gaze estimation techniques with a focus on deep learning-based classification and regression techniques.

A. MACHINE LEARNING-BASED CLASSIFICATION METHODS FOR GAZE ESTIMATION

1) 2D CLASSIFICATION-BASED METHODS FOR GAZE ESTIMATION

a: CLASSIFICATION METHODS FOR DRIVER GAZE ZONE ESTIMATION

Different studies have designed classification models for gaze estimation. Choi *et al.* [12] proposed a five-layered CNN-based technique for driver gaze zone classification and head-pose estimation. They created a dataset consisting of different images of male and female drivers, including drivers using eyeglasses. Based on the dataset, they built a CNN model that can classify 9 gaze zones of drivers and estimate their head-pose. Each gaze zone represents different regions in a car, including left mirror, right mirror, rear-view mirror, steering, gear, middle, left windscreen, and right windscreen. A version of AlexNet architecture [40] was used in the study, and experimental results showed that the technique produced a classification accuracy of 95% correct gaze zone detections. There have been significant improvements in the design of personalized driver gaze zone classification systems. Nevertheless, the design of generalized systems is still lagging [41]. Generalized systems should be capable of classifying gaze zones of different subjects and different perspectives Vora, *et al.* [41] took a step forward by designing a generalized CNN-based technique for gaze detection. The technique consists of two units, namely the pre-processing and fine-tuning units. The pre-processing unit consists of three pre-processing strategies used to extract sub-images (from the raw input images) that are most relevant for eye gaze classification. The first strategy involves

extracting the full-face image for training, while the second strategy involves extracting the upper half of the face for training. In the third strategy, some context was added to the driver's face by extending the face bounding box in different directions. The images were extracted and passed to the fine-tuning unit for training. The fine-tuning unit consists of two pre-trained models: AlexNet and VGG-16 [42]. Overall, the authors built six different CNN-based gaze classification models, based on a different combination of the above-mentioned pre-processing strategies and pre-trained model. During training, one pre-processing technique and one pre-trained model were chosen and trained at a time. The experimental result shows that the combination of VGG-16 and half face image produced the best classification accuracy of 93.36%. The combination of AlexNet and half face image produced an accuracy of 88.91%. The authors noted that the low performance of AlexNet is due to the large kernel size (11×11) and a stride of 4 used in the first convolutional layer of the AlexNet model. This large kernel size and stride cannot efficiently capture slight movements of the eyelid or pupil, which is very important because the gaze zone changes with slight movements of the eyelid or pupil. The VGG-16 model consists of convolutional layers with a kernel size of 3×3 and a stride of 1. The small kernel size and stride enabled the network to capture slight eye movements for efficient gaze estimation. In another study [39], the same authors introduced a similar gaze classification approach using four pre-trained networks, namely: AlexNet, VGG-16, ResNet50 and SqueezeNet. However, instead of using three pre-processing units and two fine-tuning units, they used four pre-processing and four fine-tuning units. They evaluated the different configurations, and the best configuration (SqueezeNet + Half face image) produced a classification accuracy of 95.18%.

Monitoring the gaze attention of drivers in automobiles poses some risks to the driver, such as extreme lighting changes, obstructions, and dizziness from long drives [3]. Some gaze estimation systems require the driver to use glasses, which can affect the accuracy of the monitoring system [3]. Naqvi, *et al.* [3] tackled this challenge by introducing a CNN-based model using a near-infrared (NIR) camera that considers head and eye movements and does not obstruct the view of drivers. The designed system is made up of one NIR camera, one zoom lens and six NIR light-emitting diodes (LEDs) for illumination. The NIR camera is used to capture the frontal view image of the driver, which is then transmitted to a laptop through a USB interface line. The Dlib facial feature tracker [43] was used to detect 68 different face landmarks, which are then used as reference points to extract a region of interest (ROI) images of the driver's face, left eye, and right eye. The mean of all the pixel values in each of the ROI images was calculated and used to normalize the brightness of the ROI images. Normalization improves the performance of learning models and simultaneously reduce the effect of light [3]. Three sets of features (for face, left eye, and right eye) were extracted by passing the captured face, left and right eye images through three different VGG-16 pre-trained models. The gaze zone of a driver was then calculated by combining the extracted features from the three networks, using a method described by.Naqvi, *et al.* [3]. The accuracy of the proposed method was measured based on two metrics: strictly correct estimation rate (SCER) and loosely correct estimation rate (LCER). More information on the two metrics is provided in Section IV. The experimental results show that it achieved an average SCER and LCER of 92.8% and 99.6%, respectively.

b: CLASSIFICATION METHODS FOR NEAR-EYE DISPLAY GAZE ESTIMATION

Some studies introduced deep learning methods for Virtual Reality (VR) systems [14], [44], [45]. VR systems attempt to imitate several natural stimuli to provide a sense of immersion into VR [14]. In every immersion experience, the major interphase between the user and the non-physical world is the near-eye display [14]. Near-eye display methods focus on predicting eye gaze on a screen placed close to the eye. Some characteristics of near-eye display that determine the quality of user experience are resolution, contrast, field of view, eye tracking, and positional tracking. Some of these characteristics have been improved in VR systems, except eye gaze tracking [14].

Many techniques have been proposed to bridge this gap; however, the vast majority are suitable for far-eye displays; they focused on predicting eye gaze on a screen placed some distance away from the eye [14]. Far-eye display methods are not suitable for near-eye displays, where eye imaging is limited to a small region [14]. This is because their performance might be affected by occlusions and a partial view of the iris and pupil [14]. Konrad, *et al.* [14] addressed this problem by introducing an end-to-end CNN-based gaze estimation technique for near-eye displays. They created a dataset containing eye images of users looking at various calibration points on a screen. The images were captured by a camera placed very close to the face of different subjects. Based on the dataset, they built a simple CNN model (using the LeNet architecture) that takes the images of users as input and estimate the gaze direction of the users, based on the x and y coordinates on the screen. The authors treated the gaze estimation problem as a multi-class classification problem, where each class is treated as a point on the screen. The technique was evaluated on the captured dataset, and it produced an angular error of 6.7 degrees, which is poor. The poor performance is likely due to the poor image quality (28×28) and the dataset variability used to train the network. The dataset contains images from only 5 subjects. The poor performance may be also due to the LeNet-based architecture used to learn the image-gaze mappings. The LeNet architecture is not large enough to extract all the features necessary for accurate gaze estimation. It was originally designed for handwritten digit recognition [46]. The technique was also evaluated on the CAVE (Columbia Gaze Data Set) [47]. The dataset contains 5880 images from 56 subjects for 21 gaze directions and 5 unique head poses. The technique was evaluated on three classes and 21 classes

and it produced a classification accuracy of less than 30% for the 21 classes and less than 95% for the three classes.

Hickson, *et al.* [45] proposed a technique for classifying facial expressions in VR systems using eye trackers. The technique is designed to automatically infer facial expressions by analysing only a partially occluded face of users while they are engaged in a VR experience. In the study, images of the user's eye are captured from a gaze tracking camera within a headset and used to infer a subset of facial expressions. The images are used to generate dynamic avatars in real-time which serves as an expressive substitute for users. They introduced a new approach for improving the accuracy of deep CNN models. The technique was evaluated, and it achieved a mean accuracy of 74%.

Garbin, *et al.* [48] introduced a large-scale dataset for training models for VR systems. The dataset consists of eye images captured using a VR head-mounted system with two synchronized eyes facing camera. The images in the dataset were captured from the eye regions collected from 152 subjects and it is divided into four subsets. The first subset consists of 12,759 labelled images, while the second subset consists of 252,690 unlabelled images. The third subset contains 91,200 frames randomly selected from video sequences of 1.5 seconds, and the last subset contains 143 pairs of left and right point cloud data. An experiment was performed to evaluate the quality of the dataset and the results show that it is suitable for building eye-tracking models for VR applications. Kim, *et al.* [44] created two datasets for near-eye gaze estimation problems. The first dataset contains over 2 million synthetic images with variations in face shape, gaze direction, skin colour, pupil, iris, and external conditions. The second dataset contains 2.5 million images collected from 35 subjects. They evaluated the quality of the dataset by training NN models, and it achieved an accuracy of 2.06 degrees.

c: CLASSIFICATION METHODS FOR GAZE-BASED TYPING

Some studies introduced gaze estimation techniques for on-screen keyboard typing. These techniques use eye blink as an input to an eye tracker. They facilitate text communication using eye movement Zhang, *et al.* [13] proposed a calibration-free appearance-based technique that allows users to enter input text by merely looking at the on-screen keyboard and blinking their eyes. They divided the human gaze into nine directions, namely: left-up, up, right-up, left, middle, right, left-down, down, and right-down. They also built a dataset consisting of several images from 25 people with different lighting conditions, eye appearance, locations, and time. The images were captured using mobile phones, webcams, and digital cameras. They implemented many data augmentation methods to improve the robustness of the model to image resolution, illumination, slight head rotation, and skin colour. They divided eye states into ten groups, consisting of the nine directions (outlined above), and one eye-closed state. Based on the dataset, they built a CNN gaze classification model that can learn the ten eye states from two eye images. The model was evaluated, and it produced an accuracy of 95.01%.

Rustagi, *et al.* [49] introduced a gaze estimation technique for touchless typing using head movement-based gestures. To type a sequence of letter, a user makes a series of gesture by looking at the desired letter on a virtual QWERTY keyboard. The sequence of gestures is captured by a face detection deep neural network-based system provided in OpenCV [50]. The processed video frames are then used as an input to a pre-trained HopeNet model [51]. The HopeNet model is a CNN-based landmark-free head pose estimation model built to compute intrinsic Euler angles (yaw, pitch, and roll) from an RGB image. The output of the pre-trained model was then used to train an RNN model that predicts the sequence of clusters a user is looking at. The predicted cluster sequence is used to suggest valid dictionary words that can be formed out of the sequence. The method was evaluated on a dataset consisting of 2234 video sequences collected from 22 subjects, and it achieved an accuracy of 91.81% Alsharif, *et al.* [52] introduced another gaze-based method for typing using the LSTM network. In the method, input data was used to train an LSTM network. During training, the Connectionist Temporal Classification (CTC) loss function was used to allow the network to output characters directly without the need of HMM states. This function allows the network to map input sequences to short output sequences. After training, the network produces a matrix that corresponds to the set of permitted characters. To constrain the output to a limited set of words, a Finite State Transducer was used. The method was evaluated, and experimental results show that it produces a classification accuracy of 92%.

d: CLASSIFICATION METHODS FOR BIOMETRIC SYSTEMS

Biometric identification techniques have attracted more attention due to the growing demands of improved security solutions. Eye-tracking can be used as an additional biometric input to improve the performance of biometric systems. Liang, *et al.* [53] introduced a video-based identification model for biometric systems using eye-tracking techniques. They created video clips for different subjects to view, with the goal of capturing eye-tracking data that reflects their physiological and behavioural attributes, such as acceleration, muscle, and geometric attributes. These attributes are extracted from the captured eye gaze data and used as biometric features to identify individuals. They used a feature selection algorithm (based on mutual information of features) to select relevant features for biometric identification. The selected features were then used to train two classifiers, namely: NN and SVM. The experimental results show that measuring video-based eye-tracking data is a practicable solution for biometric applications.

Jia, *et al.* [54] proposed a framework for biometric identification through eye movement. They used RNN to learn dynamic eye features and temporal dependencies from a dataset captured from a sequence of raw eye movement recordings. The model was designed to work in a task-independent manner by using short-term feature vectors combined with different stimuli in the training and testing

phase. They evaluated the model on a dataset containing samples of 32 subjects presented with static images. The samples were recorded using a video-based eye-tracker calibrated for each subject. Experimental results showed that the technique achieved a Rank-1 Identification Rate of 96.3% for identification scenario and an equal error rate (EER) of 0.85% for the verification scenario. Trokielewicz, *et al.* [55] proposed a CNN-based biometric technique for post-mortem iris recognition. They fine-tuned a pre-trained model that was originally trained on image segmentation task, called SegNet [56]. The fine-tuning was performed using a labelled dataset containing cadaver iris images collected from 42 subjects. Experimental results show that it achieved an EER of less than 1% for samples collected up to 10 hours after death. It also produced an EER of 21.45% for samples collected up to 369 hours after death.

2) 3D CLASSIFICATION-BASED METHODS FOR GAZE ESTIMATION

Eye-tracking techniques have high accuracy in performing gaze estimation in the x and y direction, but not in depth [57]. Some applications (such as VR/AR systems) requires accurate measurements of the x-and y-direction of the eye gaze, particularly the focal depth information. Changwani and Sarode [58] proposed a low-cost technique for VR systems using neural networks. The technique consists of two cameras for tracking the head movements in three dimensions. Input from the two cameras is used to train a neural network model for predicting the gaze location of users. Foveated rendering technique is used to improve the immersion experience of users. It is performed by rendering the area of the screen where a user is looking at and distorting the area that falls under the peripheral vision.

Shin, *et al.* [59] introduced a gaze depth estimation method for VR and AR systems. They used a binocular eye tracker to collect a wide range of features from two eyes, including pupil centre, gaze direction, and inter pupil distance. These features are then used to build a NN model for predicting gaze depth. The method was evaluated on a dataset consisting of images from 13 subjects. In the evaluation, individual models were designed for each of the 13 subjects, and a generalized model was also designed for the 13 subjects. The experimental results show that the method produced a gaze depth accuracy of 90.1% for the individual models and 89.7% for the generalized model. Lee, *et al.* [57] proposed an NN-based technique for determining gaze depth in a head-mounted eye tracker. They used a binocular eye tracker with two cameras to capture gaze information of subjects looking at fixed points at distances from 1m to 5m. The recorded gaze vectors were used to train a NN model. The model was evaluated, and the results show that it produced an average classification error of less than 10%.

3) SUMMARY

The knowledge of gaze direction can provide valuable insights into the point of attention for different users. Many other classification techniques have proposed for gaze estimation, such as techniques for identifying Eye Accessing Cues (EAC) of users. EAC refers to certain patterns of eye movements that provide information about the cognitive processes of the human brain [60], [61]. George and Routray [23] designed a computationally inexpensive real-time regression-based CNN model for estimating eye accessing cues in a desktop environment. In the study, eye images were extracted from a dataset containing face images of different subjects. The eye images were then used to build a CNN-based classification network for predicting seven EAC classes. The network was trained on the left and right eye images independently and jointly. All the experiments were performed for both 3 and 7 classes, respectively. Experimental results showed that the network that was trained on two eyes images outperformed the network trained on a single eye. The network produced a classification accuracy of 89.81%. A summary of some selected gaze estimation classification techniques is reported in Table 1. The table presents a list of some appearance-based classification techniques and their various performances. It also provides information on the architectures, datasets, and image resolution used by various techniques.

B. MACHINE LEARNING-BASED REGRESSION METHODS FOR GAZE ESTIMATION

1) 2D REGRESSION-BASED METHODS FOR GAZE ESTIMATION

a: REGRESSION METHODS COMBINED WITH HANDCRAFTED OR EXTERNAL FEATURES

Regression-based gaze estimation methods attempt to estimate a mapping function from the input variables (x) to a numeric or continuous output variable (y). Different regression-based solutions have been designed for eye gaze estimation. Krafka, *et al.* [19] introduced a regression-based eye-tracking system for mobile devices, called iTracker. They created a large-scale dataset (called GazeCapture) consisting of over 2 million images from over 1450 people using mobile phones and tablets. The dataset was collected under variable lighting conditions, different backgrounds, diverse orientations, and various head pose. The inputs to the eye-tracking system (i.e. iTracker) include full-face images, left-eye images, right-eye images, and a face grid. The face grid is a 25 $\times$ 25 binary mask input used to infer the eye and head poses for each image. Based on the dataset, the iTracker network was trained end-to-end to predict the distance, in the X and Y directions, from the camera of mobile phones and tablets. Experimental results showed that the model achieved a prediction error of 1.77 cm and 2.83 cm on mobile phones and tablet, respectively. The trained model was fine-tuned to each device and orientation, and it achieved a reduced prediction error of 1.71 cm and 2.53 cm for mobile phones and tablet, respectively. In a different study, Kim, *et al.* [10] introduced another regression-based model for gaze estimation on mobile devices. Inspired by the results

TABLE 1. Summary of appearance-based classification methods for gaze estimation.

Ref.	Accuracy	Architecture	Dataset	Image Resolution	Added Feature	Network input	Application area
[12]	95%	Own architecture	Own dataset	42 × 50	-	Full face	Automobile
[14]	6.7	LeNet	CAVE, Own dataset	28 × 28	-	Two eyes	VR/AR systems
[23]	89.81%	Own architecture	Eye Chimera	42 × 50	-	Two eyes	Desktop environments
[41]	93.36%	AlexNet, VGG16	Own dataset	224 × 224, 227 × 227	-	Full face	Automobile
[13]	95.01%	Own architecture	Own dataset	32 × 128	-	Two eyes	Gaze-based typing
[3]	92.8 – 99.6%	VGGFace16	CAVE, Own dataset	224 × 224	PCCR Vector	Full face and two eyes	Automobile
[39]	95.18%	AlexNet, VGG16, ResNet, SqueezeNet	Own dataset	224 × 224, 227 × 227	-	Full face	Automobile
[59]	89.7% - 90.1%	Own architecture	Own dataset	-	-	Two eyes	VR/AR systems
[55]	EER: 1%, 21.45%	SegNet	Own dataset	240 × 320	-	Two eyes	Forensics (post-mortem iris recognition)
[54]	EER: 0.85%	-	Own dataset	-	-	A sequence of gaze points from the two eyes	Biometrics
[53]	-	Own architecture	Own dataset	-	-	Two eyes	Biometrics

achieved by the iTracker network, they designed a model (called Gazelle) that takes five inputs, namely: right eye, left eye, face, face grid, and Histogram of Gradients (HOG). The HOG feature was created by computing the histogram of oriented gradients from the cropped face images of each frame. This was done to provide the network with an additional feature (than just pixel values) that can better capture useful qualities for gaze estimation, such as head pose. They trained the model on images from the GazeCapture dataset. They used the OpenCV pre-trained Haar cascade classifier to obtain face and eye bounding boxes from the images. However, the OpenCV classifier could not accurately detect faces and eyes from low resolution or cropped images. Therefore, only 15% (350,000 images) of the GazeCapture images could be used for training. They trained the model on the 350,000 images and it produced a prediction error of 4.85 cm, which represents the distance, in the X and Y directions, from the camera of a mobile device. The authors noted that the HOG feature improved the model's performance by 0.31 cm.

Some conventional techniques employed hand-engineered features. However, some of these methods are not very reliable under natural light conditions or free head movements [21]. Wang, *et al.* [21] addressed this problem by introducing a gaze estimation technique based on CNN and random forest regression. Instead of using hand-crafted features, they introduced a hybrid technique for learning CNN-based image features (called deep features) for gaze estimation. Specifically, they trained a CNN model on different eye images and extracted the output of the last fully connected layer of the network. The extracted CNN-based features were then used to train a random forest regressor to learn mappings between the deep features and gaze coordinates. The technique achieved a prediction error of 1.53°. The results show that the deep features substantially improve performance on regression-based algorithms compared to hand-engineered features.

b: REGRESSION METHODS WITHOUT HANDCRAFTED FEATURES

Generally, eye tracking applications for consumer devices are expected to be computationally inexpensive with low power consumption [22]. Some studies tackled this challenge by introducing hardware optimized CNN-based regression techniques.

Lemley, *et al.* [22] designed a hardware-friendly CNN model for low-cost consumer devices. The model was designed to predict a gaze angle that corresponds to low-resolution eye images. They trained the model on images from the MPIIGaze dataset [65], a standard dataset used for gaze estimation in unconstrained environments. Specifically,

they trained the model on left-eye images, right-eye images, and on both left and right eyes images. The result showed that using information from two eye images (i.e. left and right eye) can improve the accuracy of a CNN model. Besides, they evaluated the sensitivity of the model to distance, by down-sampling the evaluated images to different image resolutions and training the model on the down-sampled images. The result showed that the model's performance reduces when a subject is far from the camera, implying that the model is sensitive to distance. They also evaluated the effect of data augmentation on the model, and the result showed that data augmentation improved the model. They designed a network for hardware efficiency and improved accuracy. The network consists of four convolutional layers, two max-pooling layers, one dropout layer, and two fully connected layers. The first and second convolutional layers consist of 32 filters, while the last two convolutional layers consist of 64 filters. One of the important requirements considered in the study for designing the hardware optimized network was the kernel size. They stacked the network with kernels of sizes 3×3. The network was evaluated, and it produced a prediction error of 3.64 degrees.

Some existing CNN-based gaze estimation models are trained on datasets collected under controlled environments, implying that these models are not capable of effectively estimating gaze in real-world conditions [11]. Zhang, *et al.* [11] tackled this challenge by introducing an image feature extraction technique for eye tracking in uncontrolled environments. They created a dataset (called MPIIGaze) consisting of over 213,000 images from 15 laptop users. The dataset was used to build a CNN-based model for appearance-based gaze estimation in unconstrained environments, such as driver gaze monitoring systems. They used a face detection and facial landmark detection technique to detect face and facial landmarks from different images in the dataset. A generic 3D facial shape model was also used to estimate 3D head poses of the detected faces. The head poses, and eye images were used to train a CNN network that learns mappings from the head poses and eye images to different gaze directions. The technique was evaluated on the MPII Gaze dataset, and it produced a prediction error of 13.9 degrees for cross-dataset evaluation and 6.3 degrees for within-dataset evaluation. The prediction error was further reduced when the model was trained on subject-specific information. It produced an error of $\sim$3 degrees.

Zhang, *et al.* [18] postulated that other face regions (beyond the eye) contain important information for gaze estimation. To prove this, they designed a spatial weight mechanism that can competently encode different regions of the entire face into a CNN network. Full face images were captured and passed through several convolutional layers to generate a resultant feature map. The feature map was then passed through the spatial weight mechanism to generate a weight map. Afterwards, the weight map was multiplied with the feature map using element-wise multiplication. Finally, the output map was fed into fully connected layers to produce the estimated gaze values. The technique was evaluated, and it produced an angular error of 4.8 degrees and 6.0 degrees for MPIIGaze and EYEDIAP datasets, respectively.

2) 3D REGRESSION-BASED METHODS FOR GAZE ESTIMATION

a: REGRESSION METHODS FOR FREE HEAD 3D GAZE TRACKING

Some of the existing gaze estimation techniques are only capable of estimating gaze positions on a screen [8], [66], but do not provide information on gaze vector and eye location. These techniques did not capture the relationship between head pose, eyeball movement, and gaze vectors [62]. They were simply designed to learn this relationship from the dataset, which will lead to overfitting of the head-gaze relationship [62]. Some techniques achieved state-of-the-art performance; however, these techniques could only predict gaze intersection on a screen and not on a 3D surface [62]. Some applications need information on the specific object or region a user is looking at [62]. These applications require gaze estimation techniques that can calculate the gaze intersections between the 3D gaze vector and various objects in a 3D scene [62]. Examples of such applications include driver gaze attention monitoring, analysis of advertisement and investigations [62]. Free-head 3D gaze estimation techniques can estimate both the eye location and the gaze vector in a 3D space [63]. Zhu and Deng [62] proposed an effective and low-cost 3D gaze estimation technique for eye tracking. They introduced a strategy for building gaze estimation models that can effectively capture both free head movements and eyeball movements. In the strategy, they divided the gaze estimation into two separate modelling tasks, where they trained two different CNN models for modelling the eye movement and head movements. They introduced an additional layer (called gaze transform layer) that combines the predictions from the two CNN models, and aggregate them into a gaze vector. They also proposed a two-step training strategy that divides the training task into two stages. In the first stage, the eyeball and head pose model were trained separately on coarse head pose and eyeball movement labels. In the second stage, the two models were trained jointly on accurate gaze labels. The authors created a dataset consisting of 240,000 images from 200 subjects with full coverage of various head poses, eyeball movements, lighting conditions, glasses occlusions and reflections. The technique was evaluated, and it achieved a cross-subject error of 4.3°.

Extrapolating human gaze from low-resolution eye images is still a challenging task. Sugano, *et al.* [63] addressed this problem by proposing a learning-by-synthesis technique for appearance-based gaze estimation. They used a fully calibrated multi-camera system to collect a large amount of cross-subject 3D annotated datasets consisting of 64,000 images from 50 subjects. They also constructed a synthesized dataset by performing a 3D reconstruction of the eye regions in the collected dataset. Using the synthesized dataset, they trained

TABLE 2. Summary of appearance-based regression methods for gaze estimation.

Ref	Accuracy	Architecture	Dataset	Image Resolution	Added Features	Network input	Application area
[19]	~2 cm, about 3	iTracker	Gaze Capture	224 × 224	Face grid	Full face and two eyes	Mobile phones and tablets
[10]	4.85 cm	iTracker	Gaze Capture	144 × 144	HOG	Full face and two eyes	Mobile phones
[22]	4.63	Own architectures	MPII Gaze, UT Multiview	-	-	Two eyes	Consumer electronic systems
[21]	1.53	DeepID	Own dataset	40 × 70	-	One eye	In the wild environments
[18]	4.8 , 6.0	AlexNet	MPII Gaze, EYEDIAP	448 × 448	Spatial weights	Full face	2D and 3D gaze estimation
[38]	29.53 mm	iTracker	Own Dataset	400 × 120	-	Face	
[1]	1.0 , 1.3	Own Architecture	MIT	100 × 100	-	Fixation patch	Indoor environments
[11]	13.9	Own Architecture	MPIIGaze	36 × 60	-	Full face	In the wild environments
[62]	4.3	AlexNet	MPIIGaze, UT Multiview	224 × 224	Gaze transform layer	Full face and one eye	3D gaze tracking
[63]	6.5	Own architecture	Own dataset	15 × 9	-	Synthetized eye images	3D gaze tracking
[32]	10	-	Own dataset		-	Two eyes	3D gaze tracking
[64]	5.1 and 6.2	VGG-16	EYEDIAP	128 × 48	Facial landmarks	Full face and two eyes	3D gaze tracking

a random forest regression model for 3D gaze estimation, and it produced a cross-subject mean error of 6.5 degrees.

Funes Mora and Odobez [32] introduced another method for gaze estimation in 3D surfaces. They used a 3D Morphable Model to generate person-specific 3D templates for different faces in a dataset. Specifically, they used the Basel Face Model [67], which was built from a large group of subjects. Using the person-specific templates, they estimate the head pose for each subject using a 3D face tracker. The eye tracker is based on video and ground truth data obtained from a multimodal Microsoft Kinect sensor. Based on the estimated head pose and 3D templates, they map the head image to a frontal image of the face and crop the eye region from the resulting image. The eye images are then used to train an adaptive linear regression model for estimating the eye gaze in the reference system of the head. The eye gaze is then transformed to a gaze direction in the world coordinate system. The method was evaluated, and it produced a mean error greater than 10 degrees.

Palmero, *et al.* [64] tackled the problem of head-pose and person-independent 3D gaze estimation in remote camera. They designed a method that can model both appearance-based and shape-based gaze estimation cues. The appearance-based cues are represented by the full-face and eye images, while the shape-based cues are represented by 3D facial landmarks obtained from a 68-landmark model which models the global shape of the face. The appearance-based and shape-based features are used to jointly train an RCNN model for 3D gaze estimation. The method was evaluated for both static and moving head scenarios. The experimental results show that it produced an average angular error of 5.1 and 6.2 degrees for the static and moving head, respectively.

3) SUMMARY

Many other appearance-based regression gaze estimation methods have been proposed in the literature. Table 2 presents a summary of some selected regression-based gaze estimation techniques.

IV. GENERAL DISCUSSION

This section presents a general discussion on CNN-based gaze estimation techniques. The section is divided into four subsections. The first subsection provides a discussion on existing gaze estimation techniques, while the second subsection provides a discussion on the comparison between calibration-based and CNN-based techniques. The third subsection provides a discussion on various datasets used to evaluate the performance of CNN models, and the last subsection provides information on the network parameters used to train CNN-based gaze estimation models.

A. MACHINE LEARNING METHODS FOR EYE TRACKING

Eye-tracking is a very interesting domain that has attracted the interest of many researchers. It is a great tool for any human behaviour research applied in different domains,

including psychology, medicine, marketing, and automobile [3], [68]. It can also be used to improve the interaction between humans and computers, as eye gaze can be used for navigation and control. A good number of eye gaze estimation techniques have been proposed and applied in the literature [20]. In recent times, however, we have experienced a departure from conventional eye gaze tracking methods - the majority of which are model-based methods. Compared to model-based approaches, much work has not been done on appearance-based eye gaze estimation. Some of the existing appearance-based methods addressed 2D and 3D gaze estimation. Earlier studies focused on ANNs, random forest, linear regression, SVRs, deep end-to-end CNN models, and transfer learning. However, compared to other appearance-based methods, the CNN-based gaze estimation approach has not been fully explored. Zhang, *et al.* [11] are one of the first authors to apply CNNs to eye gaze estimation.

Eye-tracking can be applied in constrained and unconstrained environments. Constrained gaze tracking systems are not very useful for some eye-tracking problems, especially problems that involve free head movements, such as driver gaze monitoring. Such gaze detection systems will not be effective if users move their eye to gaze at something without moving their head. Therefore, some studies focused on designing improved CNN models for unconstrained environments, such as automobiles, mobile devices, and laptop devices. As shown in Table 1, CNNs have not been fully explored for driver gaze classification. Among the few existing methods, some of them were designed to handle seven gaze zones [39], [41], while some were designed to handle six gaze zones [12]. These methods are not very reliable as they cannot accurately determine driver positions. Given this, Naqvi, *et al.* [3] designed an improved method for handling 17 gaze zones, and it outperformed the previous techniques with smaller gaze zones. This implies that performance increases with an increase in the number of gaze zones.

Different testing methods can be used to evaluate gaze estimation models. Some studies used the cross-dataset testing method to evaluate their models [39], while some studies used within-dataset testing method [12]. The cross-dataset testing method ensures that the images in the training datasets are completely different from the images in the test datasets. This is the case in a real-world scenario, where the trained system can be used by different subjects without personalizing the system for each subject. Choi, *et al.* [12] did not use the cross-dataset testing method, but they used the within-dataset testing method. In their study, 70% frames for each subject were used for training, and the other 30% frames were used for testing and validation. Vora, *et al.* [39] replicated the experimental setup used by Choi, *et al.* [12] and reported that they achieved a high classification accuracy of 98.7%. The same authors [39] used a cross-dataset testing method for the same dataset and obtained a lower classification accuracy of 82.5%. This shows that the within-dataset testing method is not reliable because it overfits to subject-specific features. The method does not reflect the true generalization ability of a designed model. Developers are advised to be very careful when choosing evaluation techniques for deep learning gaze estimation models.

The task of developing a deep learning model for gaze estimation can be considered as a classification or regression task. Although both considerations are useful, regression provides the best classification flexibility [22]. Gaze classification can be broadly divided into outdoor environment and indoor desktop environments [3]. Indoor desktop environments can be further divided into wearable and non-wearable device-based techniques. Wearable device-based techniques consist of a camera and illuminator mounted on the head of the subject in the form of a pair of glasses or a helmet [69]–[71]. Although wearable device-based techniques can accommodate free head movements, they are not very convenient, especially when users are required to wear the devices for long durations. Therefore, non-wearable device-based techniques were introduced to tackle this problem [72], [73]. These techniques use non-wearable gaze tracking devices (such as illuminators and RGB cameras) to acquire face and eye images for gaze estimation. A typical example of non-wearable device-based techniques is the pupil centre corneal reflection (PCR) method introduced in [74], [75]. These techniques can effectively handle-free head movements. They can also accommodate free eye movements. They are very suitable for gaze estimation, as they do not need complex geometric knowledge on lighting, cameras or eyes [3]. Non-wearable device-based techniques are more convenient than wearable device-based techniques, but they require initial user or camera calibration which can be burdensome and time-consuming. Some authors proposed a calibration-free PCR-based technique for gaze estimation. Experimental results show that PCR-based methods are negatively affected by badly captured images, where the pupil and cornea reflection are not properly detected. Camera calibration is also required for PCR-based gaze estimation in outdoor environments, which can be cumbersome.

In deep learning, fine-tuning of pre-trained CNN models is a common practice and it has generally improved performance. Through fine-tuning of pre-trained CNN models, reduced prediction error can be achieved with almost negligible time to train the dense layers. Fine-tuning can be performed on models that have been pre-trained on generic datasets. The goal of pre-training is to extract general image structure, while the goal of fine-tuning is to extract domain-specific features. As shown in Tables 2 and 2, network architectures can be fine-tuned for gaze estimation, including AlexNet, VGG-16, LeNet, ResNet, SqueezeNet, and iTracker. Fine-tuning can be achieved by updating the pre-existing weights of all the network layers or replacing the final layer with a new layer that is trained from scratch. The former is used when the output dimensionality is similar to the dimensionality of the new domain, while the latter can be used when the output dimensionality is different from the dimensionality of the new domain [38].

The performances of the different pre-trained models vary with different gaze estimation tasks. For example, Vora, *et al.* [41] achieved a gaze classification accuracy of 93.36% when they used the pre-trained VGG-16 model, but they achieved a lower classification accuracy of 88.91% when they used the pre-trained AlexNet model. This is because VGG-16 was pre-trained for a task closely related to eye gaze estimation (face recognition), compared to AlexNet model that was pre-trained for object recognition. The network depth of VGG-16 is also larger than the network depth of AlexNet, which further explains its better performance. Nevertheless, AlexNet architecture (pre-trained on ImageNet) is still vastly used by many gaze estimation studies because the early layers of pre-trained AlexNet have already learned different features like curves and edges, which are very useful to most classification problems, such as eye gaze classification, face recognition, and image classification. It is worthy to note that using a small learning rate during fine-tuning is recommended, primarily to avoid losing the weights already learned by previous layers of the network. We can use one pre-trained model for fine-tuning, or a combination of pre-trained models, as shown in [39], [41].

The size of the training datasets plays an important role in the accuracy of gaze estimation models. Generally, CNN models produce very good results when trained on a large volume of datasets. The results reported in the literature show that the prediction error decreases significantly as the sample size increases. This implies that small datasets are not very suitable for building efficient CNN models from scratch. Small datasets are mostly useful for fine-tuning models. In the event where we have a meagre amount of data, data augmentation can be used to increase the gaze accuracy and generalization ability of CNN models. Some data augmentation techniques include flip, rotation, and scale. In addition to dataset size, variations in data samples also play an important role in achieving good performance. Models that are trained on a large volume of datasets of the same class will not produce very good generalization performance. Datasets are required to contain images of different classes, appearances, head pose, and illuminations. This is to ensure that CNN models capture these variations and provide reliable gaze estimation. Some studies introduced large-scale datasets for eye gaze estimation, such as MPIIGaze, EYEDIAP, and GazeCapture. These datasets contain images with varying head poses, illumination, and facial expressions.

Calibration in CNN models is not necessary for building efficient gaze estimation models. Krafka, *et al.* [19] studied the effect of calibration by simulating the process of calibration. In the study, images from 13 fixed positions and 47 random positions were collected from each subject. The images from the fixed positions were used to fine-tune a pre-trained CNN model, and the remaining images were used to evaluate the model. The model was fine-tuned on a different number of calibration points, and results showed that its performance decreases when trained on a few calibration points, which could be due to overfitting. The model produced its best performance when trained on 13 calibration points. Specifically, the model achieved an improved prediction error of 1.34 cm and 2.12 cm, compared to 1.71 cm and 2.53 cm that was achieved without calibration. Although calibration is useful, its impact on CNN models is not very significant because of the generalization ability of CNN models, achieved through deep learning and large-scale datasets.

A different number of inputs can be used to train CNN frameworks. Some frameworks can be trained on multiple inputs independently, while some can be trained on single inputs. Naqvi, *et al.* [3], Kim, *et al.* [10] designed a framework that consists of three deep CNNs, where each CNN takes the left eye, right eye and full-face image as input, respectively. Results reported in the literature show that each of the three inputs contributes to the overall performance of CNN models [19]. However, the contribution of each input varies. The full-face input provides the highest contribution, compared to the left and right eye. This suggests that a more efficient technique can be designed using a framework that takes the full-face image as input. Some studies designed CNN models that accept hand-engineered features as inputs. For example, Krafka, *et al.* [19] introduced an eye grid as an external feature, while Kim, *et al.* [10] introduced HOG. The eye grid is a binary feature that specifies the size and location of the head inside the frame, while HOG is a semantically-rich feature that provides information on some important gaze estimation features of the face, such as head pose. Results reported in the literature show that hand-engineered features can reduce the prediction errors of CNN gaze estimation models [10].

CNNs can be applied to inputs of different dimensions, however, due to their success in 2D images, they are mostly applied to 2D inputs [22]. CNNs have been applied to other inputs including 1D inputs (time series) and 3D inputs (volumetric or time series data) Wang, *et al.* [1] designed 2D and 3D regression-based methods using Regression-based Convolutional Neural Network (RCNN) and model-based gaze estimation method. Specifically, they used RCNNs to learn image features that correlate well with eye fixations. Eye images of users were captured and used as input to RCNNs for estimation of fixation map. The fixation map was then combined with a model-based method to obtain the gaze prediction. Results reveal that the 2D calibration method produced a prediction error of 1.00 degrees, while the 3D method produced a prediction error of 1.4 degrees.

The quality of images used to train various gaze estimation models varies; some studies used high-resolution images, while some used low-resolution images. Results reported by Choi, *et al.* [12] show that CNNs are capable of reliably processing images of low resolution, unlike other model-based approaches that require high-resolution images for accurate gaze prediction. On the contrary, the results reported by Konrad, *et al.* [14] shows that low-quality images can negatively affect the classification accuracy of CNN-based eye-tracking systems. Konrad, *et al.* [14] trained their model on images of size 28×28, and it produced a very poor classification accuracy of 0.003%. Image resolution should

TABLE 3. Comparison between calibration-based and CNN-based techniques.

Ref	Accuracy	Configuration	Head movement
CNN-based techniques			
[19]	~2 cm	Eye and Face Images, Face grid	Yes
[22]	4.63	Eye images	Yes
[21]	1.53	Eye images	Yes
[18]	4.8 , 6.0	Image dataset	Yes
[1]	1.0 , 1.3	Image dataset	Yes
[12]	95%	Full face images	Yes
[62]	4.3	Face and eye images	Yes
[41]	93.36%	Face images	Yes
[13]	95.01%	Eye images	Yes
[3]	92.8 – 99.6%	Eye and face images	Yes
[39]	95.18%	Face images	Yes
Calibration-based techniques			
[81]	1.25	2 cameras, 2 light sources	Yes
[76]	3.78	1 Kinect sensor + integrated LEDs	Yes
[82]	0.6	4 Cameras, 2 LEDs, stereo systems	Yes
[83]	1.0	1 camera, stereo system, and LED	Yes
[84]	0.87	1 camera, 1 infrared light source	No
[85]	1.5	2 video cameras	Yes
[86]	1.11	2 IR light sources, 1 camera	No
[77]	1.3	4 LEDs and webcam	Yes
[87]	1.4	1 video camera	No

be taken into careful consideration when building CNN-based gaze estimation models.

B. CALIBRATION-BASED GAZE ESTIMATION VS DEEP LEARNING-BASED GAZE ESTIMATION

This section presents a comparison between calibration-based and deep learning-based gaze estimation techniques. As shown in Table 3, calibration-based techniques produced better results compared to calibration-free techniques. Calibration-based models are most suitable for applications that do not require subject-independence or pose-independence. This is because, some of them assume a fixed head pose [18], implying that they do not permit free head movement. Most of them also require subject-specific training, while some of them are evaluated on a few subjects which limit their generalization performance [76], [77]. Some of them also require complex calibration procedures [78], such as screen calibration and personal calibration. Some

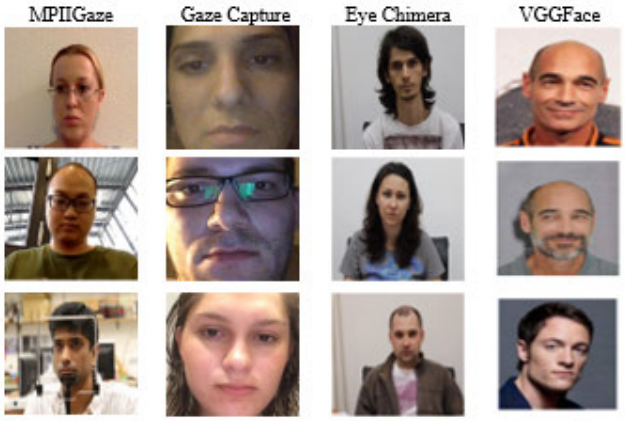


FIGURE 5. Sample images from four datasets.

of them require very high-resolution images which can be computationally expensive to process, while some of them cannot be applied to multiple devices and orientations.

In contrast, deep learning-based gaze estimation models can handle both pose-independent and subject-independent gaze estimation and do not require complex setups or calibration procedures. They have higher generalization performance (compared to calibration-based techniques) because they are trained on a large number of images from many subjects ranging from 50+ subjects [47], [62], [63], [79] to over 1000 subjects [19], [80]. They are useful for unconstrained eye tracking because they permit free head movements. They are also generalizable to multiple devices and orientations.

Some studies designed deep learning-based gaze estimation systems that estimate 2D (X, Y) gaze coordinates relative to the camera [10], [19]. The X coordinate specifies the distance (in cm or mm) to the left or right direction of the camera on a virtual plane that contains the true location, while the Y coordinate indicates the distance in the up and down direction of the camera. This coordinate system allows the model to predict gaze coordinates that is generalizable to multiple devices (such as tablets, laptops, and smartphones) and orientations (portrait or landscape, depending on the placement of the camera on the screen). It leverages the fact that the front-facing camera is usually on the same plane to the screen, and it is angled perpendicular to the screen [19]. Different devices are typically used to record the gaze tracking data for building CNN models, including smartphones, tablets, and laptops. For each recording, dots are randomly displayed on the screen of these devices and users are asked to fixate at each dot location. The 2D gaze coordinates for the dot locations are recorded, and in some cases converted to a corresponding dot location on a normalized prediction space. The screen size measurement for each device and the placement of their camera are also recorded. The recorded gaze data is then used to build gaze estimation models.

One of the challenges of deep learning-based gaze estimation models is the lack of balanced and variable large-scale

TABLE 4. Some selected datasets for eye gaze estimation.

Ref.	DATASET	Year	Size	Head poses	Gaze direction	Identities	Purpose
[103]	[103]	2007	2,200	19	2 - 9	20	Eye gaze and head pose
[97]	LFW	2008	13233	Continuous	-	5,749	Eye gaze and head pose estimation
[104]	PUT	2008	9971	5	-	100	Face recognition
[105]	HPEG	2009	20 videos	2	-	10	Evaluation of eye gaze and head pose estimation.
[106]	RS-DMV	2010	10 videos	-	-	-	Eye gaze and head pose estimation
[98]	YouTube Faces	2011	3425 videos	Continuous	-	1595	Face recognition
[102]	WDRef	2012	99773	NA	NA	2,995	Face recognition
[107]	Celebface	2013	87628	Continuous	NA	5, 436	Gaze estimation
[47]	CAVE-DB	2013	5880	5	105	56	Face recognition
[79]	[79]	2013	1236	1	12	103	Eye gaze estimation
[108] [109]	Eye Chimera	2013	1170	-	7	40	Eye gaze estimation
[110]	Microsoft COCO	2014	328124	NA	NA	91	Eye gaze estimation
[101]	Celebfaces+	2014	202599	Continuous	NA	10, 177	Image classification
[111]	Facebook	2014	4.4 million	Continuous	-	4,030	Face recognition
[112]	Eyediap	2014	videos	Continuous	Continuous	16	Face recognition
[63]	UT Multiview	2014	64000	8+synthesized	160	50	Eye gaze estimation
[92]	VGG-Face	2015	2.6 million	Continuous	NA	2622	Appearance-based gaze estimation.
[99]	TabletGaze	2015	816 videos	4	35	51	Face recognition
[93]	Google	2015	200 million	Continuous	-	8 million	Unconstrained gaze estimation
[11]	MPIIGaze	2015	213659	Continuous	Continuous	15	Evaluation of gaze tracking techniques
[96]	UnityEye	2016	1 million	Continuous	Continuous	-	Appearance-based gaze estimation in the wild
[19]	GazeCapture	2016	2445504	Continuous	13+ Continuous	1474	Gaze estimation
[3]	DDGC-DB1	2018	39108	Continuous	17	20	Driver gaze classification
[44]	Nvgaze	2019	2 million	Continuous	-	35	Near-eye gaze estimation
[44]	Nvgaze	2019	2.5 million synthetic images		-	-	Near-eye gaze estimation

dataset. Most of the available datasets are not well representative of devices and orientations. The images in these datasets were captured from multiple devices, but the variability and number of training data from each device (represented in the dataset) are not well balanced. To achieve very good performance, each device should be well represented, in terms of size and fixation coverage. The training data from each device should cover a wide range of fixation locations for that device. The training data from each device should also have a good representation of variability in appearance, head pose, facial expression, illumination, and orientation.

C. DATASET FOR GAZE ESTIMATION

The introduction of deep learning in 1943 [88] has meaningfully improved many applications in the computer vision domain [89]. Evidence from previous studies [90], [91] shows that one secret to this success is the availability of large-scale datasets. In the past, researchers in the domain of computer vision were faced with the challenge of accessing large-scale public datasets, and because of this, many of the advances in this domain remained restricted to internet giants, such as Facebook, Twitter, and Google [92]. For example, in 2015, Google released a face recognition technology that was trained on 200 million images from eight million individuals [93]. However, in recent times, many researchers are collecting and making datasets available for different computer vision tasks. Parkhi, *et al.* [92] introduced a large-scale dataset for face recognition tasks consisting of over 2 million images from over 2600 identities. Many other datasets have been introduced in the literature. Figure 5 shows the sample images from some datasets used in the literature. Some of these datasets contain images with different sizes, backgrounds, illuminations, and appearances. Some datasets also contain head pose data (such as head motions) so that appearance-based models can learn free head movements. Generally, datasets are expected to have a wide variety in the head pose, illumination, appearance, and background. Although there are standard benchmark datasets

TABLE 5. Some selected parameters for training deep learning-based gaze estimation models.

Ref.	MOMENTUM	Initial learning rate	Weight decay	Mini-batch size	Epoch	Optimizer
[19]	0.9	0.001, reduced to 0.0001 after 75,000 iterations.	0.0005	256	150,000	-
[3]	0.9	0.000001	0.0005	20	16	SGD
[10]	0.9	0.00001	0.9^{epoch}	20	10	SGD
[41]	-	0.0001	-	32 (AlexNet) & 64 (VGG16)	5	Adam
[39]	-	4×10^{-4} (for SqueezeNet) and 0.0001 (for AlexNet, VGG16 and ResNet50)	-	64 (AlexNet & SqueezeNet), 32 (VGG16) & 64 ResNet50)	50	Gradient Descent
[42]	0.9	0.001, decreased 3 times	0.0005	256	74	SGD
[113]	0.9	0.1, divided by 10 when the error plateaus.	0.0001	256	60×10^4	SGD
[111]	0.9	0.01, manually decreased by order of magnitude as soon as the validation error stops decreasing. Decreased finally to 0.0001.	-	128	15	SGD

used for image classification (such as ImageNet [94] and MNIST [95]), there is no dataset that has become the standard benchmark for gaze estimation tasks [38].

Some early studies constructed appearance-based models with small datasets. These models, however, do not generalize well to gaze estimation in-the-wild. Gaze estimation in-the-wild is characterized by variability in illumination, eye appearance, and head pose. Recent appearance-based models, therefore, trained models on large-scale datasets. For example, the GazeCapture dataset contains over 1.4 million usable images collected from over 1400 users with different head poses, appearance, and illumination conditions. MPIIGaze dataset contains over 213,000 images from 15 people looking at different gaze positions. The dataset was collected over three months during daily laptop usage. Some of these datasets are still limited in head pose and illuminations. Collecting such large-scale datasets can be very laborious, time-consuming, and expensive. Instead of training models on real eye images, some studies introduced methods for generating synthesized eye images. Wood, *et al.* [96] introduced a novel framework that combines a large amount of eye region images for training. In the study, one million synthesized images were generated using a combination of 3D eye models and a real-time rendering framework. The quality and suitability of the dataset were assessed by comparing its performance to other datasets with real images, and it achieved competitive results.

Some useful benchmark datasets for gaze estimation include VGG-Face [92], MPIIGaze [11], Labeled Faces in the Wild dataset (LFW) [97] and YouTube Faces (YTF) [98], TabletGaze [99], TurkerGaze [100], Celebfaces [101], BayesianFace [102]. It is noteworthy to mention that ImageNet dataset [94] does not contain face images; however, as shown in various results reported in the literature, fine-tuning an ImageNet-based model produces improved results for eye gaze estimation. This is because an ImageNet-based model has already captured different features like curves and edges (in its early layers), which are very useful to most classification problems. Table 4 presents a list of some useful datasets for eye gaze estimation. Although the scope of this study is eye gaze estimation, we have also included popular datasets used for face recognition, because results reported in the literature show that they are also very useful for eye gaze estimation.

D. NETWORK PARAMETERS USED FOR TRAINING APPEARANCE-BASED GAZE ESTIMATION MODELS

As shown in Table 5, the following parameters are required for training a CNN: momentum, number of iterations, mini-batch size, learning rate, and weight decay. The value for each parameter is selected based on the problem solved. We observed that Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 works well for most studies. Momentum helps to increase the speed of the gradient vector in the right directions, therefore leading to faster convergence. SGD optimizer is one of the popular optimization algorithms, and numerous state-of-the-art models are trained with it. Different learning rates can be used for training, depending on the model. Learning rate controls the frequency at which the network weights are adjusted for the gradient loss. Some studies started with equal learning rate for all the trainable layers and manually decreased the rate by a certain order of magnitude after some conditions. It is particularly

TABLE 6. Some useful deep learning architectures that can be fine-tuned for gaze estimation.

Architecture	YEAR	Layers	Input dim	Outputs	Details
AlexNet [40]	2012	1In, 5 Conv, 3Pooling, 2FC	224 × 224	1000-way softmax	Trained on 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest. A variant of the model was evaluated in the ILSVRC-2012 competition it produced a test error rate of 15.3%.
VGG [42]	2014	1In, 16 conv, 5Pooling, 3FC	224 × 224	1000-way softmax	Trained on 1.3 million images, validated on 50K images and tested on 100K images, in the ILSVRC-2012 dataset.
DeepID2 [114]	2014	4Conv, 3Pooling, 1FC	55 × 47	160-way softmax	Extracted 400 face patches from face images in CelebFaces+ dataset and trained 200 deep CNNs. Evaluated on LFW dataset and achieved 99.15% face verification accuracy.
Yaniv et al. [111]	2014	1In, 2Conv, 2Fc, 3 Locally connected layers.	152 × 152	N-way softmax, where N is the number of classes. The objective is to maximize the probability of the correct class (i.e., the face id).	Trained on 4 million face images from 4000 Facebook users. It achieved a classification accuracy of 97.35% on LFW dataset.
GoogLeNet [115]	2015	1In, 60Conv, 15Pooling, 5FC	224 × 224	1000-way softmax	Trained on 1.2 million images, validated on 50,000 images and tested on 100,000 images from the ILSVRC 2014 dataset.
Zhang et al. [11]	2015	1In, 4Conv, 2Pooling, 1FC	36 × 60	2-way regression. Head angle vector is combined with the output of the fully connected layer.	Trained on MPIIGaze dataset consisting of 213,659 images from 15 subjects
Deep Face [92]	2015	1In, 13Conv, 5Pooling, 3FC	224 × 224	2622-way softmax and L-dimension metric embedding, where L=1024	Trained on a dataset consisting of over 2.6 million face images from over 2600 subjects. Evaluated on LFW and YTF datasets and it achieved excellent results.
LeNet [14]	2016	1In, 2 Conv, 2 Pooling, 1FC	42 × 50	10	Trained on a dataset that consists of 7316 calibration points.
ResNet [113]	2016	1In, 33Conv, 2Pool, 1FC	224 × 224	1000-way softmax	Residual learning framework to ease network training. Trained on ImageNet dataset and achieved 3.57% error.
SqueezeNet [116]	2016	1In, 2Conv, 8Fire Modules, 4Pooling	224 × 224	1000-way softmax	Evaluated on ImageNet and achieved similar accuracy with AlexNet with 50X fewer parameters.
iTracker [19]	2016	4In, 4Conv, 3FC, 1Out	224 × 224	2-way Euclidean loss	Trained on 1,251,983 images, validated on 179, 496 images and tested on 59,480 images.
Anjith and Aurobinda [23]	2016	1In, 3Conv, 3Pooling, 1FC	42 × 50	7-way softmax	Trained on the Eye Chimera Database consisting of 1170 images from 40 identities.
Yafei et al. [21]	2016	1In, 4Conv, 3Pooling, 1FC	40 × 70	N-way softmax, where N is the number of calibration points.	Trained on 107,681 images collected from 6 subjects. The dataset contains images with 41 calibration points (25 points used for training and 16 used for testing).
Kang et al. [1]	2016	1In, 3Conv, 3Pooling, 2FC	100 × 100	Linear regression is used to calculate the probability of a patch being a fixation patch.	Trained on 500 images, tested on 503 images in the MIT1003 dataset. MIT dataset consists of 1003 landscape and portrait images of different objects like cars, people

TABLE 6. *(Continued.)* **Some useful deep learning architectures that can be fine-tuned for gaze estimation.**

					and faces.
David [38]	2017	1In, 5Conv, 3Pooling, 1FC	400 × 200	The network output 252 features which are then transformed into a tensor and fed into different architectures based on the output class of the architecture. The final layer consists of one feature of size 24 × 24, which corresponds to the gaze probability heatmap.	Trained on a dataset consisting of 1,917,742 bright and dark pupil ROI images. The dataset was split into training and test sets with a 9:1 ration on subject level.
Haoping and Wangjiang [62]	2017	2In, 5Conv, 3Pooling, 2FC	72 × 72	2-way Euclidean loss	Trained on a dataset consisting of 200 identities.
Chi et al. [13]	2018	1In, 5Conv, 3Pooling, 1FC	32 × 128	10-way softmax	Trained on eye images from 25 identities. Dataset consists of 832 training images, 728 validation images, and 935 test images. The training set was further transformed and expanded to 74101 eye images.
Rizwan et al. [3]	2018	4In, 13Conv, 5Pooling, 2FC	224 × 224	17-way softmax	Initialized on pre-trained VGG16 dataset and fine-tuned on DDGC-DB1 dataset.
Matthew et al. [10]	-	5In, 4Conv, 2Pooling, 3FC	144 × 144	2-way Euclidean loss	Trained on 350,000 images from GazeCapture dataset.

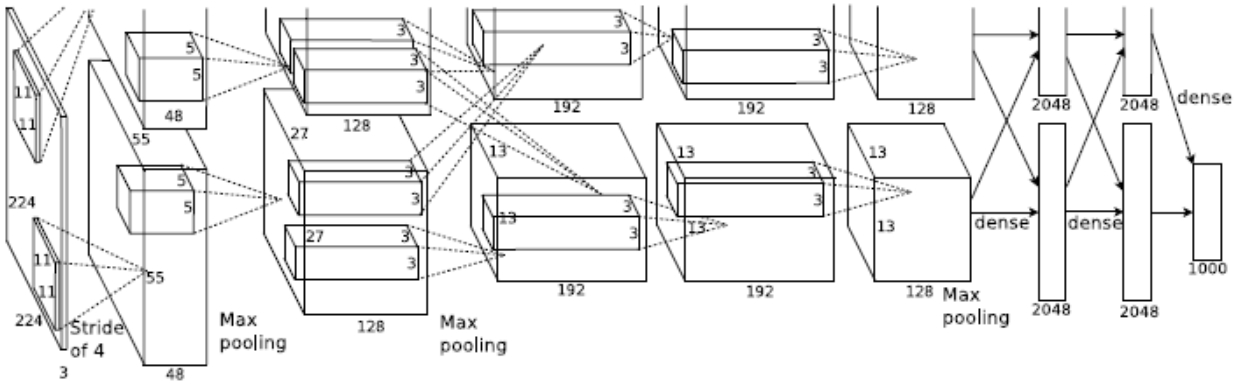


FIGURE 6. AlexNet network architecture [40].

recommended that a smaller learning rate is used when fine-tuning a model, to ensure that the pre-trained weights are preserved. Table 6 shows a description of some useful deep learning network architectures that can be fine-tuned for eye gaze estimation. Figures 6 - 12 also shows a representation of some deep learning architectures, including popular architectures, such as SqueezeNet, DeepID2, DeepFace, VGG-16, AlexNet, LeeNet, and iTracker. These architectures can be adopted and used to design deep learning-based gaze estimation techniques.

V. LIMITATIONS, FUTURE WORK DIRECTION, AND SUMMARY

This section outlines the limitations of the appearance-based gaze estimation techniques reviewed in this study. It also provides future research directions and the summary of the survey.

A. LIMITATION AND FUTURE WORK DIRECTIONS

Experimental results reported in the literature [1] shows that some of the proposed gaze estimation techniques produced state-of-the-art results. However, every study has limitations, which are normally the basis for future research. This section provides some limitations of CNN-based gaze estimation techniques.

a. CNN-based techniques can be negatively affected in cases when there are severe head and eye rotation during image capture. This can cause one of the two eyes to disappear in the captured image and consequently cause an increased error in gaze estimation. This limitation can

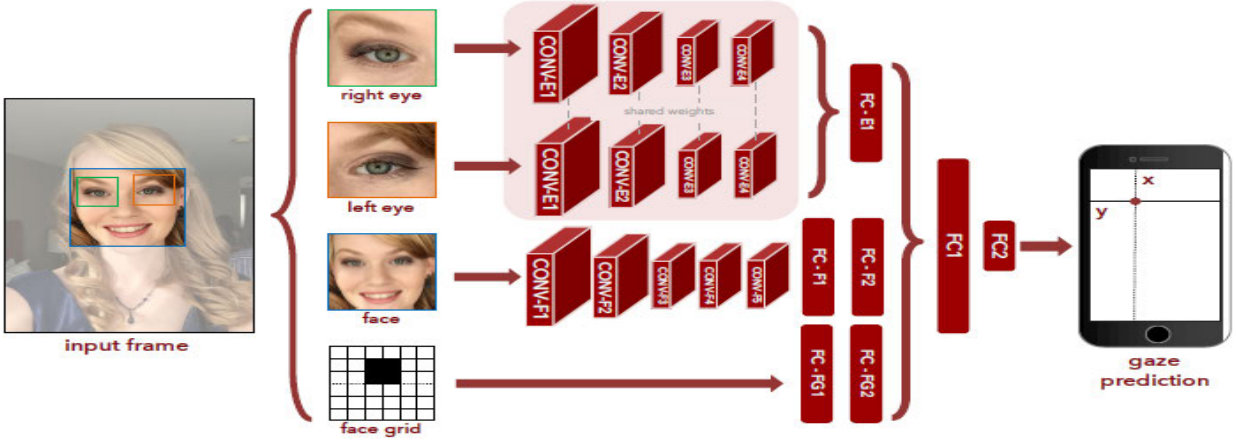


FIGURE 7. iTracker architecture [19].

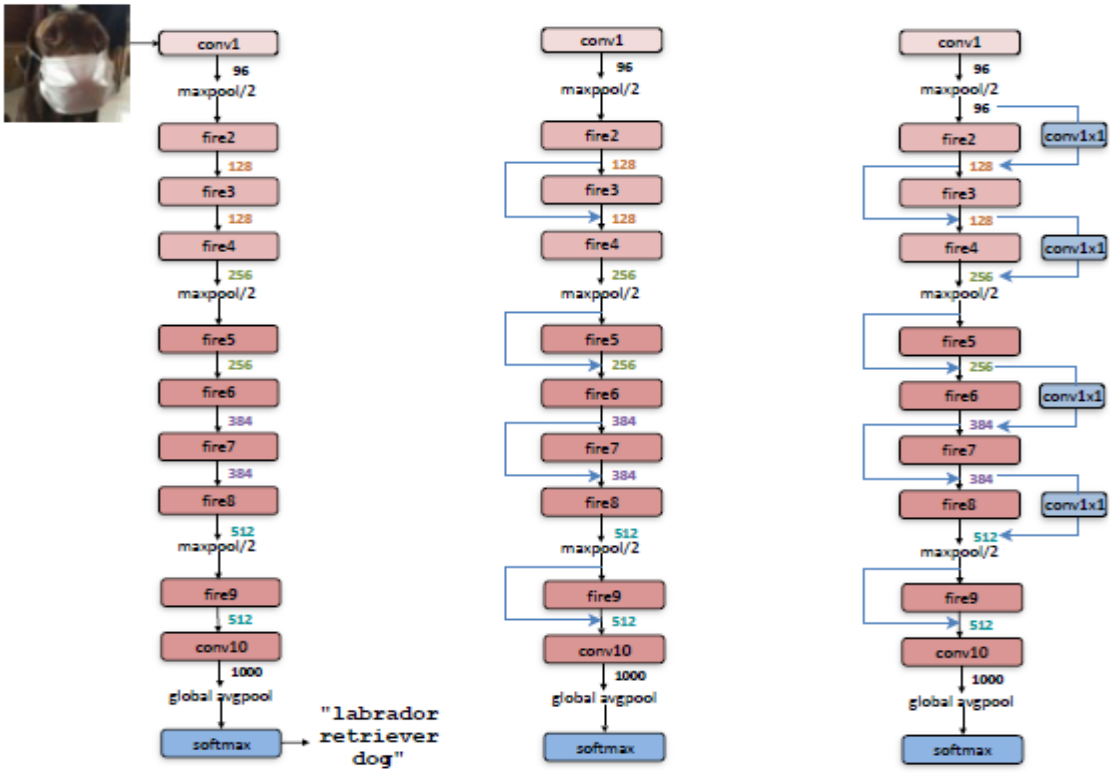


FIGURE 8. SqueezeNet architecture [116].

be resolved by using multiple cameras, which could also increase the processing time. Therefore, future research can focus on designing fast and improved gaze estimation methods that can efficiently handle free head movements and eye rotations.

b. As shown in Table 3, calibration-based gaze estimation techniques produced better prediction error than CNN-based techniques. Most of the CNN-based gaze estimation techniques produced a prediction error that is greater than 0.5°. One of the primary reasons could be due to the lack of balanced and variable large-scale dataset. Most of the available datasets are not well representative of devices and orientations. The images in these datasets were captured from multiple devices, but the variability and number of training data from each device (represented in the dataset) are not well balanced. To achieve very good performance, each device should be well represented, in terms of size and fixation coverage. The training data from each device should cover a wide range of fixation locations for that device. The

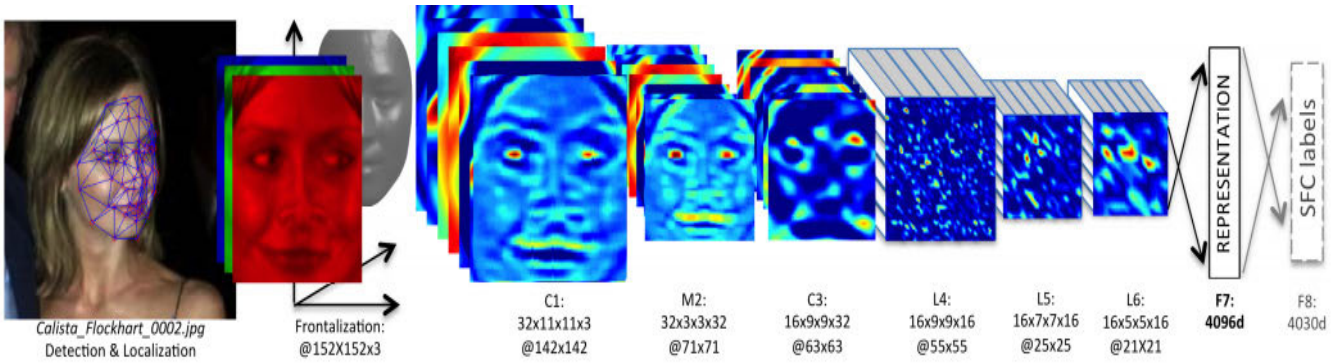


FIGURE 9. DeepFace architecture [111].

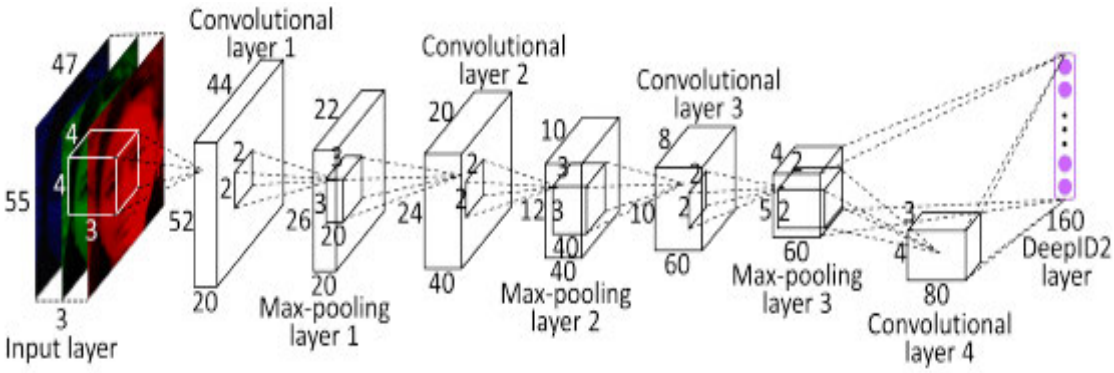


FIGURE 10. DeepID2 architecture [114].

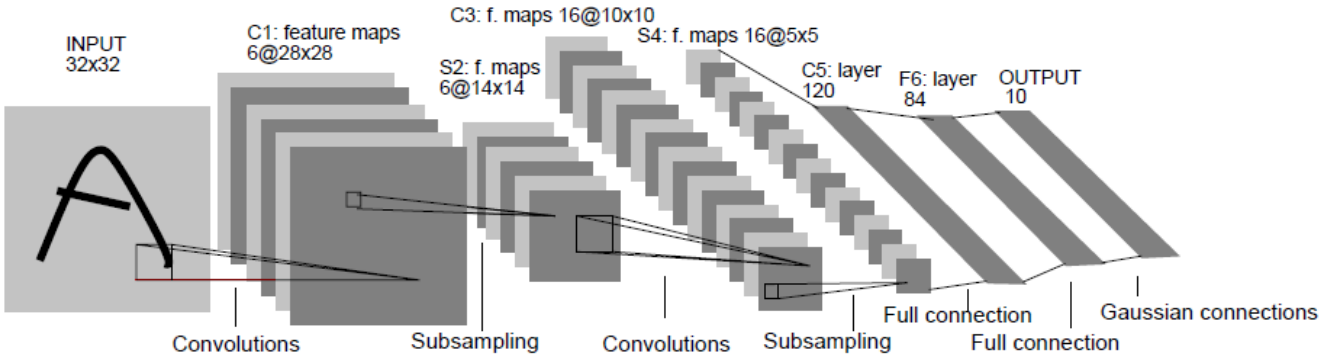


FIGURE 11. LeNet-5 architecture [117].

training data from each device should also have a good representation of variability in appearance, head pose, facial expression, illumination, and orientation. Future work can focus on collecting balanced large-scale dataset for calibration-free gaze estimation.

c. Based on some observations reported in the literature, designing robust and effective CNN-based models can be very time-consuming, costly to implement and computationally expensive. They are time-consuming because CNN performs many computationally expensive operations that require developers to have access to very fast computers, such as cluster computers with GPU accelerations or fast multi-core CPUs. CNNs are also costly to implement, especially in cases where developers do not have access to computers with large memory and storage space. Robust and accurate CNN models require well-balanced large-scale datasets for reliable eye gaze estimation. Future research can focus on designing low-cost, computationally inexpensive, hardware friendly and hardware optimized network architectures for eye gaze estimation.

d. Some studies [18] proposed heatmap gaze estimation models instead of the popular 2D gaze estimation models. Experimental results reported in the literature

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

FIGURE 12. VGG-16 network architecture [42].

suggests that heatmap models perform worse than 2D gaze point models, which could be because of their sensitivity to hyper-parameter values or difficulty to train [38]. Future studies can consider developing improved heatmap models for gaze estimations.

e. Some datasets do not contain complete information needed for constructing accurate gaze estimation models. For example, 30% of the frames in the GazeCapture dataset were not properly captured; they do not have both face and eye detections. Some of the frames are out of focus or cropped in such a way that the face of the user is occluded. These limitations can affect image classifiers (such as OpenCV Cascade classifier) that have a poor success rate in detecting eye and faces from poor quality images. Having less noise in datasets could improve the gaze accuracy of CNN models. Future research should focus on capturing well-balanced and noise-free datasets with variation in illumination, head poses, facial expressions, and eye appearances.

f. Zhang, *et al.* [13] designed a CNN-based text entry system capable of estimating eye gaze. The method is suitable for traditional 9-key T9 input mode widely used by candy bar phones. These brands of phones are becoming obsolete; hence the method may not be fully beneficial to most mobile phone users. Current dispensation eye gaze estimation techniques should be designed for the popular keyboard layout, also known as QWERTY keyboard layout.

g. Most of the CNN-based techniques reviewed in this article are designed for unconstrained environments, where captured images can be subjected to different head poses, appearance, illumination, facial occlusions, and more. Their results show that there is still much room for improvement. Future research can target designing robust and improved unconstrained techniques that can handle many real-world variations.

h. Models pre-trained on full-face images inspired the technique introduced by Masko (2017). However, the same technique was fine-tuned on ROI images. The difference in the full-face model and ROI image model could lead to variations that can negatively affect the gaze accuracy of the final model. Ideally, to achieve good results, models that are designed for ROI images should use models that are pre-trained on ROI images and not full-face images.

i. Although training CNNs on low-quality images improves the training speed, results reported in the literature [14] shows that low-quality images significantly affect the gaze accuracy of CNN-based eye gaze estimators. Future studies should consider developing techniques with balanced speed-accuracy trade-off.

j. Some of the gaze estimation models proposed in the literature were trained on relatively small datasets [108]; hence they did not produce competitive results. Generally, well-balanced, and large-scale datasets are required to construct effective and robust deep learning models. This should be the focus of future studies.

k. Appearance-based gaze estimation methods are capable of implicitly modelling the eye geometry and eye features from an input dataset, without explicit calibration. However, they are not capable of efficiently modelling variations in head positions. This is because the eye appearance may appear similar under different head poses and gaze directions. Changes in illumination (under the same pose) can change the appearance of the eye and affect the gaze estimation accuracy [8]. Future research can design improved techniques that can handle this problem.

l. As shown in Table 1, some studies introduced gaze estimation techniques for automobiles. Most of them did not consider drivers with low vision. Drivers with low vision require a bioptic lens system for driving purposes. A bioptic lens system is a system that combines two-lens optical system with one or more telescopes attached to the lens of a pair of eyeglasses [118]. Future work may consider designing techniques that can handle gaze estimation for drivers with low vision.

B. SUMMARY

This article presents a survey of appearance-based gaze estimation methods, with a focus on CNNs. Gaze estimation methods can be divided into model-based or appearance-based techniques. Model-based techniques can be further divided into corneal reflection and shape-based methods. They perform gaze estimation by finding the location of the eye on a 3D space using reflections in the eye, also called glints [119]. Corneal reflection-based methods rely on external light sources for feature detection, while shape-based methods rely on the observed shape of the eye for gaze estimation, such as iris edges and pupil centres. Unfortunately, these methods are not capable of accurately handling images of low quality or images with variable light conditions. Appearance-based methods rely on the appearance of the eye for gaze estimation. They directly map image features to gaze points without the use of hand-engineered features. Unlike model-based approaches, they can reliably handle images of low resolution. They can also generalize well on new faces without using specific data from users. Early appearance-based models introduced gaze estimation techniques that presumed a fixed head pose and training data for users. Later works introduced improved methods that can handle variable head pose, illuminations, and appearance. However, these methods still require the models to be trained on subject-specific features. Recent studies [11], [18] are focusing on introducing methods that can handle variable head pose and subject-independent gaze estimation. However, some of these methods require a huge amount of training datasets. To satisfy this need, different studies [11], [112] introduced large-scale gaze estimation datasets consisting of images with different head pose and illumination conditions.

This article focused on deep learning algorithms (and other ML algorithms) because of their popularity and efficacy. Deep learning algorithms are successful thanks to the availability of large-scale datasets and scalable computational resources such as thousands of CPU cores and GPUs. The primary goal of this study is to provide the research community with a comprehensive reference point suitable for enhancing the design of state-of-the-art appearance-based eye gaze estimation techniques. We hope that this study will be useful to the research community and will foster the design of improved gaze estimation techniques.

REFERENCES

[1] K. Wang, S. Wang, and Q. Ji, "Deep eye fixation map learning for calibration-free eye gaze tracking," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl. (ETRA)*, 2016, pp. 47–55.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[3] R. Naqvi, M. Arsalan, G. Batchuluun, H. Yoon, and K. Park, "Deep learning-based gaze detection system for automobile drivers using a NIR camera sensor," *Sensors*, vol. 18, no. 2, p. 456, Feb. 2018.

[4] S. Robertson, G. Penn, and Y. Wang, "Exploring spectro-temporal features in end-to-end convolutional neural networks," 2019, *arXiv:1901.00072*. [Online]. Available: http://arxiv.org/abs/1901.00072

[5] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2741–2749.

[6] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368–2378, May 2018.

[7] A. Tsukada, M. Shino, M. Devyver, and T. Kanade, "Illumination-free gaze estimation method for first-person vision wearable device," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 2084–2091.

[8] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 478–500, Mar. 2010.

[9] H. R. Chennamma and X. Yuan, "A survey on eye-gaze tracking techniques," 2013, *arXiv:1312.6410*. [Online]. Available: http://arxiv.org/abs/1312.6410

[10] M. Kim, O. Wang, and N. Ng, "Convolutional neural network architectures for gaze estimation on mobile devices," Standford Univ., Standford, CA, USA, Tech. Rep., 2016.

[11] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4511–4520.

[12] I.-H. Choi, Y. G. Kim, and T. B. H. Tran, "Real-time categorization of driver's gaze zone and head pose -using the convolutional neural network," in *Proc. HCI Korea*, Jan. 2016, pp. 417–422.

[13] C. Zhang, R. Yao, and J. Cai, "Efficient eye typing with 9-direction gaze estimation," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19679–19696, Aug. 2018.

[14] R. Konrad, S. Shrestha, and P. Varma, "Near-eye display gaze tracking via convolutional neural networks," Standford Univ., Standford, CA, USA, Tech. Rep., 2016.

[15] E. Lindén, J. Sjöstrand, and A. Proutiere, "Learning to personalize in appearance-based gaze tracking," 2018, *arXiv:1807.00664*. [Online]. Available: http://arxiv.org/abs/1807.00664

[16] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2033–2046, Oct. 2014.

[17] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "Appearance-based gaze estimation with online calibration from mouse operations," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 6, pp. 750–760, Dec. 2015.

[18] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 51–60.

[19] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2176–2184.

[20] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[21] Y. Wang, T. Shen, G. Yuan, J. Bian, and X. Fu, "Appearance-based gaze estimation using deep features and random forest regression," *Knowl.-Based Syst.*, vol. 110, pp. 293–301, Oct. 2016.

[22] J. Lemley, A. Kar, A. Drimbarean, and P. Corcoran, "Efficient CNN implementation for eye-gaze estimation on low-power/low-quality consumer imaging systems," 2018, *arXiv:1806.10890*. [Online]. Available: http://arxiv.org/abs/1806.10890

[23] A. George and A. Routray, "Real-time eye gaze direction classification using convolutional neural network," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jun. 2016, pp. 1–5.

[24] A. Kar and P. Corcoran, "Performance evaluation strategies for eye gaze estimation systems with quantitative metrics and visualizations," *Sensors*, vol. 18, no. 9, p. 3151, Sep. 2018.

[25] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[26] T. Huff, N. Mahabadi, and P. Tadi, *Neuroanatomy, Visual Cortex*. Treasure Island, FL, USA: StatPearls Publishing, 2020.

[27] Stanford. (2017). *CS231n: Convolutional Neural Networks for Visual Recognition*. Accessed: Feb. 18, 2019. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[28] A. Payan and G. Montana, "Predicting Alzheimer's disease: A neuroimaging study with 3D convolutional neural networks," 2015, pp. 1–10, *arXiv:1502.02506*. [Online]. Available: http://arxiv.org/abs/1502.02506

[29] P. Skalski. (2019). *Gentle Dive Into Math Behind Convolutional Neural Networks*. Accessed: Jul. 6, 2020. [Online]. Available: https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

[30] B. Ramsundar and R. B. Zadeh, *TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[31] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Math. Comput. Simul.*, vol. 177, pp. 232–243, Nov. 2020.

[32] K. A. F. Mora and J.-M. Odobez, "Gaze estimation from multimodal kinect data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 25–30.

[33] C.-C. Lai, Y.-T. Chen, K.-W. Chen, S.-C. Chen, S.-W. Shih, and Y.-P. Hung, "Appearance-based gaze tracking with free head movement," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1869–1873.

[34] L.-Q. Xu, D. Machin, and P. Sheppard, "A novel approach to real-time non-intrusive gaze finding," in *Proc. Brit. Mach. Vis. Conf.*, 1998, pp. 1–10.

[35] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 753–760.

[36] K. A. F. Mora and J.-M. Odobez, "Person independent 3D gaze estimation from remote RGB-D cameras," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2787–2791.

[37] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 656–667.

[38] D. Masko, "Calibration in eye tracking using transfer learning," M.S. thesis, Dept. Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2017.

[39] S. Vora, A. Rangesh, and M. M. Trivedi, "Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis," 2018, *arXiv:1802.02690*. [Online]. Available: http://arxiv.org/abs/1802.02690

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[41] S. Vora, A. Rangesh, and M. M. Trivedi, "On generalizing driver gaze zone estimation using convolutional neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 849–854.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[43] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.

[44] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "NVGaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–12.

[45] S. Hickson, N. Dufour, A. Sud, V. Kwatra, and I. Essa, "Eyemotion: Classifying facial expressions in VR using eye-tracking cameras," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1626–1635.

[46] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[47] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, 2013, pp. 271–280.

[48] S. J. Garbin, O. Komogortsev, R. Cavin, G. Hughes, Y. Shen, I. Schuetz, and S. S. Talathi, "Dataset for eye tracking on a virtual reality platform," in *Proc. Symp. Eye Tracking Res. Appl.*, Jun. 2020, pp. 1–10.

[49] S. Rustagi, A. Garg, P. R. Anand, R. Kumar, Y. Kumar, and R. R. Shah, "Touchless typing using head movement-based gestures," 2020, *arXiv:2001.09134*. [Online]. Available: http://arxiv.org/abs/2001.09134

[50] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, pp. 120–125, 2000.

[51] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2074–2083.

[52] O. Alsharif, T. Ouyang, F. Beaufays, S. Zhai, T. Breuel, and J. Schalkwyk, "Long short term memory neural network for keyboard gesture decoding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2076–2080.

[53] Z. Liang, F. Tan, and Z. Chi, "Video-based biometric identification using eye tracking technique," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2012, pp. 728–733.

[54] S. Jia, D. H. Koh, A. Seccia, P. Antonenko, R. Lamb, A. Keil, M. Schneps, and M. Pomplun, "Biometric recognition through eye movements using a recurrent neural network," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Nov. 2018, pp. 57–64.

[55] M. Trokielewicz, A. Czajka, and P. Maciejewicz, "Post-mortem iris recognition with deep-learning-based image segmentation," *Image Vis. Comput.*, vol. 94, Feb. 2020, Art. no. 103866.

[56] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[57] Y. Lee, C. Shin, A. Plopski, Y. Itoh, T. Piumsomboon, A. Dey, G. Lee, S. Kim, and M. Billinghurst, "Estimating gaze depth using multi-layer perceptron," in *Proc. Int. Symp. Ubiquitous Virtual Reality (ISUVR)*, Jun. 2017, pp. 26–29.

[58] A. Changwani and D. T. Sarode, "Low-cost eye tracking for foveated rendering using machine learning," in *Proc. IEEE Int. Conf. Cloud Comput. Emerg. Markets (CCEM)*, Sep. 2019, pp. 32–39.

[59] C. Shin, G. Lee, Y. Kim, J. Hong, S.-H. Hong, H. Kang, and Y. Lee "Evaluation of gaze depth estimation using a wearable binocular eye tracker and machine learning," *J. Korea Comput. Graph. Soc.*, vol. 24, no. 1, pp. 19–26, 2018.

[60] L. L. Di Stasi, R. Renner, A. Catena, J. J. Cañas, B. M. Velichkovsky, and S. Pannasch, "Towards a driver fatigue test based on the saccadic main sequence: A partial validation by subjective report data," *Transp. Res. C, Emerg. Technol.*, vol. 21, no. 1, pp. 122–133, Apr. 2012.

[61] S. D. Goldinger and M. H. Papesh, "Pupil dilation reflects the creation and retrieval of memories," *Current Directions Psychol. Sci.*, vol. 21, no. 2, pp. 90–95, Apr. 2012.

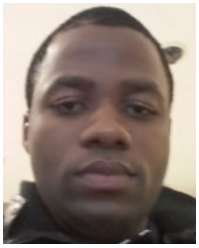
[62] H. Deng and W. Zhu, "Monocular free-head 3D gaze tracking with deep learning and geometry constraints," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3143–3152.

[63] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1821–1828.

[64] C. Palmero, J. Selva, M. A. Bagheri, and S. Escalera, "Recurrent CNN for 3D gaze estimation using appearance and shape cues," 2018, *arXiv:1805.03064*. [Online]. Available: http://arxiv.org/abs/1805.03064

[65] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 162–175, Jan. 2019.
[66] F. Onur and F. Vilariño, "Low cost eye tracking: The current panorama," *Comput. Intell. Neurosci.*, vol. 2016, no. 5, pp. 2–14, Mar. 2016.
[67] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 296–301.
[68] J. Niemann and C. Fussenecker, "Analysis of eye tracking usage in different domains and possible applications in the engineering environment," in *Proc. Int. Conf. Prod. Res.-Afr., Eur. Middle East, 3rd Int. Conf. Qual. Innov. Eng. Manage.*, Cluj-Napoca, Romania, 2014, pp. 1–6.
[69] A. Galante and P. Menezes, "A gaze-based interaction system for people with cerebral palsy," *Procedia Technol.*, vol. 5, pp. 895–902, Jan. 2012.
[70] J. M. Franchak, K. S. Kretch, K. C. Soska, and K. E. Adolph, "Head-mounted eye tracking: A new method to describe infant looking," *Child Develop.*, vol. 82, no. 6, pp. 1738–1750, Nov. 2011.
[71] B. Noris, J.-B. Keller, and A. Billard, "A wearable gaze tracking system for children in unconstrained environments," *Comput. Vis. Image Understand.*, vol. 115, no. 4, pp. 476–486, Apr. 2011.
[72] H. Lee, W. Lee, C. Cho, S. Gwon, K. Park, H. Lee, and J. Cha, "Remote gaze tracking system on a large display," *Sensors*, vol. 13, no. 10, pp. 13439–13463, Oct. 2013.
[73] J. J. Magee, M. Betke, J. Gips, M. R. Scott, and B. N. Waber, "A human–computer interface using symmetry between eyes to detect gaze direction," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1248–1261, Nov. 2008.
[74] D. H. Yoo and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," *Comput. Vis. Image Understand.*, vol. 98, no. 1, pp. 25–51, Apr. 2005.
[75] S.-W. Shih and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 234–245, Feb. 2004.
[76] X. Zhou, H. Cai, Z. Shao, H. Yu, and H. Liu, "3D eye model-based gaze estimation from a depth sensor," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2016, pp. 369–374.
[77] C. Ma, K.-A. Choi, B.-D. Choi, and S.-J. Ko, "Robust remote gaze estimation method based on multiple geometric transforms," *Opt. Eng.*, vol. 54, no. 8, Aug. 2015, Art. no. 083103.
[78] L. Sun, Z. Liu, and M.-T. Sun, "Real time gaze estimation with a consumer depth camera," *Inf. Sci.*, vol. 320, pp. 346–360, Nov. 2015.
[79] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza, "Hybrid method based on topography for robust detection of iris center and eye corners," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 9, no. 4, p. 25, 2013.
[80] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The CAS-PEAL large-scale Chinese face database and baseline evaluations," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 149–161, Jan. 2008.
[81] C.-C. Lai, S.-W. Shih, and Y.-P. Hung, "Hybrid method for 3-D gaze tracking using glint and contour features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 24–37, Jan. 2015.
[82] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2003, p. II-451.
[83] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky, "Real-time stereo tracking for head pose and gaze estimation," in *Proc. 4th IEEE Int. Conf. Automat. Face Gesture Recognit.*, Mar. 2000, pp. 122–128.
[84] P. Blignaut, "Mapping the pupil-glint vector to gaze coordinates in a simple video-based eye tracker," *J. Eye Movement Res.*, vol. 7, no. 1, pp. 1–11, 2013.
[85] Z. Zhu, Q. Ji, and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 1132–1135.
[86] Y.-G. Shin, K.-A. Choi, S.-T. Kim, C.-H. Yoo, and S.-J. Ko, "A novel 2-D mapping-based remote eye gaze tracking method using two IR light sources," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2015, pp. 190–191.
[87] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proc. 5th IEEE Int. Conf. Automat. Face Gesture Recognit.*, May 2002, pp. 131–136.
[88] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943.
[89] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, Feb. 2018, Art. no. 7068349.
[90] S.-H. Zhong, Y. Liu, and K. A. Hua, "Field effect deep networks for image recognition with incomplete data," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4, pp. 1–22, Aug. 2016.
[91] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.
[92] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, p. 6.
[93] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[94] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[95] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[96] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl. (ETRA)*, 2016, pp. 131–138.
[97] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008, pp. 1–15.
[98] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 529–534.
[99] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "TabletGaze: Unconstrained appearance-based gaze estimation in mobile tablets," 2015, *arXiv:1508.01244*. [Online]. Available: http://arxiv.org/abs/1508.01244
[100] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "TurkerGaze: Crowdsourcing saliency with webcam based eye tracking," 2015, *arXiv:1504.06755*. [Online]. Available: http://arxiv.org/abs/1504.06755
[101] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.
[102] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 566–579.
[103] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann, "A comprehensive head pose and gaze database," in *Proc. 3rd IET Int. Conf. Intell. Environ. (IE)*, 2007, pp. 455–458.
[104] A. Kasinski, A. Florek, and A. Schmidt, "The PUT face database," *Image Process. Commun.*, vol. 13, nos. 3–4, pp. 59–64, 2008.
[105] S. Asteriadis, D. Soufleros, K. Karpouzis, and S. Kollias, "A natural head pose and eye gaze dataset," in *Proc. Int. Workshop Affect.-Aware Virtual Agents Social Robots*, 2009, pp. 1–4.
[106] J. Nuevo, L. M. Bergasa, and P. Jiménez, "RSMAT: Robust simultaneous modeling and tracking," *Pattern Recognit. Lett.*, vol. 31, no. 16, pp. 2455–2463, Dec. 2010.
[107] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1489–1496.
[108] L. Florea, C. Florea, R. Vrânceanu, and C. Vertan, "Can your eyes tell me how you think? A gaze directed estimation of the mental activity," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–11.
[109] R. Vrânceanu, C. Florea, L. Florea, and C. Vertan, "NLP EAC recognition by component separation in the eye region," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2013, pp. 225–232.
[110] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[111] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.
[112] K. A. F. Mora, F. Monay, and J.-M. Odobez, "EYEDIAP: A database for the development and evaluation of gaze estimation algorithms from RGB and RGB-D cameras," in *Proc. Symp. Eye Tracking Res. Appl. (ETRA)*, 2014, pp. 255–258.

[113] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2016, pp. 770–778.
[114] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[115] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[116] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: https://arxiv.org/abs/1602.07360
[117] Y. LeCun. (2013). *LeNet-5, Convolutional Neural Network*. Accessed: Feb. 12, 2019. [Online]. Available: http://yann.lecun.com/exdb/lenet/
[118] VisionAware. (2020). *Driving With Low Vision*. Accessed: Jul. 28, 2020. [Online]. Available: https://visionaware.org/everyday-living/transportation/driving/
[119] A. T. Duchowski, *Eye Tracking Methodology: Theory and Practice*, vol. 328. Berlin, Germany: Springer, 2007, p. 614.



ANDRONICUS A. AKINYELU received the B.Sc. degree (Hons.) in computer science from the Federal University of Technology, Akure, Nigeria, in 2011, and the M.Sc. and Ph.D. degrees in computer science from the University of Kwa-Zulu Natal, South Africa, in 2015 and 2017, respectively. He is currently a Postdoctoral Research Fellow with the University of the Free State, Bloemfontein, South Africa. His areas of research interest include deep learning, machine learning, big data analytics, artificial intelligence, and computer vision.

PIETER BLIGNAUT received the Ph.D. degree in computer science from the University of the Free State. He is currently affiliated as a Researcher with the Department of Computer Science and Informatics, University of the Free State. He specializes in the technical aspects of eye-tracking and has published articles on algorithms to detect events from raw eye-tracking data, analyze the quality of eye-tracking data and visualize eye-tracking data. He is also a Regular Reviewer of eye-tracking articles and serves also as the Associate Chief Editor for the *Journal of Eye Movement Research*.