CUSTOMER RETENTION FOR E-COMMERCE USING GAZE TRACKING

Dhruv Rohatgi
School of Computing
Newcastle University
Newcastle Upon Tyne
d.rohatgi2@newcastle.ac.uk.com

Supervisor: Prof. Boguslaw Obara
School of Computing and Biosciences Institute
Newcastle University
Newcastle Upon Tyne
Boguslaw.Obara@newcastle.ac.uk

Abstract

Gaze estimation becomes an essential tool in many domains since it indicates where a person is looking; hence it is a valuable source for understanding human intention and gaze direction where the person is looking. The recent progress in deep learning algorithms has dramatically improved the performance of many computer vision tasks like gaze estimation. However, there is a lack of proper guidelines for designing deep learning algorithms for gaze estimation. Also, there's a lack of practical implementation that discourage researchers to work in this direction. This article presents a comprehensive review of the current state-of-the-art gaze estimation techniques and also proposes solution that focuses on CNN and machine learning-based gaze estimation techniques. This study aims to provide valuable insights and empower the research community to help design and develop efficient gaze estimation models.
This article also provides information on various pre-trained models, network architectures, and open-source datasets to train deep learning models. We summarize different techniques from four perspectives: feature extraction, deep learning algorithm architecture design, subject personal calibration, and device and platform used. To compare the performance of various gaze estimation approaches, we characterize all the publicly available gaze estimation datasets and present an overview of gaze estimation algorithms in tabular form. This paper not only serves as a reference to develop deep learning-based gaze estimation methods but also a guideline for future gaze estimation research. At last, a review is presented for the research on eye gaze estimation applications across many domains, including human-computer-interaction, psychology, computer vision, Marketing research, Product packaging design, website design, and Product advertisement.

***K*eywords** Gaze Tracking · Computer Vision · CNN · Landmark Detection

Dedication

I dedicate my dissertation work to my parents, whose words of encouragement and encouragement to perseverance still ring in my ears. They have never left my side and are extremely special to me. They were there for me every step of the way. I will always be grateful for everything they have done, especially for teaching me through their invaluable life experiences.

Acknowledgement

This work was completed at Newcastle University's Department of Computer Science. Throughout the dissertation process, I am grateful to my supervisor, Professor Boguslaw Obara, for his unmatched patience, encyclopaedic knowledge, and unwavering support. I could not have imagined a smooth journey without his wise mentorship. Dr. Sujit Roy and Dr. Gaurav Garg at BrainAlive Research Pvt. Ltd. for their unending support, which was crucial to my growth. Your guidance and constructive criticism helped me expand the scope of my research.
I am appreciative of my peer's valuable contributions. Mr. Nilansh Khurana and Ms. Aayushi Aggarwal are to be commended for sharing their knowledge of computer vision and its paradigm. I am grateful that they were always available for conversation and willing to answer questions. Special thanks to Brainalive Research and HelixSmartLabs for assisting me in acquiring new skills. Rishav Verma, Shreya Sureka, and Dev Munjal deserve special recognition for ensuring that I do not feel pressured.
My spiritual mentors, the late Mr. Ashwani Kumar and Mrs. Shobha Rohatgi, have taught me to dream big. Baba , I miss your hand on my head more than anything else. My parents, Mr. Mohit Rohatgi and Mrs. Jyoti Rohatgi, have demonstrated through their actions that hardwork is the essence of life. I owe my accomplishments to you. Thank you for your unrelenting efforts to mould me. I also owe Ms. Aayushi a lot for all the motivation, love, and support she gave me during my master's and before. Without her, it would not have been possible. My sister and brother-in-law, Mrs. Priyanka Rastogi and Mr. Raghav Rastogi, for always lifting my spirits and cheering me on. Dr. Suresh Kumar and Dr. Umesh Dutta, who have been my mentors, have given me praise, encouragement, and faith in every choice I've made. I have their unwavering support on every journey I've undertaken. My friends Simran Choudhary and Tushar Garg for all the late-night banter, sharing my successes and failures equally and taking pride in my achievements.

Contents

1 Introduction and problem statement **1**

2 Eye Gaze Tracking: Review **2**

2.1 Introduction . . . 2

2.2 GAZE TRACKING FUNDAMENTALS . . . 2

2.2.1 Types of eye movements . . . 2

2.2.2 Basic method and setup used for eye gaze estimation: . . . 3

2.2.3 Estimation of gaze tracking accuracy . . . 3

2.3 Eye Gaze Estimation Methodology . . . 4

2.3.1 Feature Extraction: . . . 5

2.3.2 Gaze Estimation Methodology: . . . 5

2.3.3 Calibration . . . 7

3 Data Prepration and Preprocessing **8**

3.1 Data Collection . . . 8

3.2 Calibration Window . . . 8

3.3 Data Preprocessing . . . 8

3.4 Lighting Normalization . . . 9

4 Implementation and Discussion **9**

4.1 Experiment . . . 9

4.1.1 Left and Right Eye . . . 10

4.1.2 Face . . . 10

4.1.3 Grid . . . 10

4.1.4 Fully Connected Layer . . . 10

4.2 Training . . . 11

4.3 Evaluation and Execution . . . 11

4.3.1 Evaluation . . . 11

4.3.2 Execution . . . 11

4.3.3 Calibration Process . . . 11

5 Customer Retention-Application **11**

5.1 Output . . . 12

6 Conclusion and Future Work **12**

List of Figures

1 Optical axis joins pupil and corneal lens centres. The visual axis connects the fovea and corneal lens. The point of gaze is the visual axis-screen junction[1] 4
2 Face Landmarks using dlib 68 Facial Landmarks Detector 5
3 Active calibration method 8
4 Folder Structure 9
5 Smoothing the input image and eliminating pixels with RGB values above 90 removed specularity. Images were filled with neighbouring pixel values using whole filing[2] 9
6 Gaze Tracking Training Architecture 10
7 Loss Graph 11
8 Calibration Screen 12

List of Tables

1 Characteristics of different eye movements 2

1 Introduction and problem statement

Eye gaze is considered one of the most fundamental passive forms of communication. Improvements in eye gaze tracking technology have led to the evolution of effective gaze estimation techniques for human-computer interaction over the past few decades. It all started with skin electrodes placed around the eyes for gaze prediction. With improvement and research, head-mounted eye trackers came into the market. With more focus on improving accuracy and reducing constraints for the user, post-2000 due to rapid advancement in computer processing speed, several gaze estimation methods are proposed, such as 2D regression model-based method, 3D eye model-based method, appearance-based method. 2D model-based method directly maps the feature vector to the point of gaze (POG) using the transformation function. The 3D model-based method constructs a geometric model of the eyes to estimate gaze. Both 2D regression model and 3D model requires dedicated complex setup such as infrared cameras. Appearance-based methods are non-PCCR methods that directly learn the mapping from input images to the point of gaze. Convolutional neural network-based architectures for gaze estimation are becoming very popular with the recent evolution of deep learning algorithms. These deep learning models can directly map input features to gaze direction without requiring any external calibration or with very few calibration steps compared to other methods available.
The appearance-based gaze estimation methods doesn't need a complicated setup, it only uses a webcam to capture human eye appearance. The following structure is required to estimate the gaze direction: 1) An effective feature extractor that extracts different eye features from raw facial images. 2) A robust mapping function to learn the mapping from appearance to human gaze. 3) A large sample data set to train regression function. In recent years, rapid development in deep learning algorithms has proven to be very effective in estimating good results compared to conventional appearance-based methods. It has many advantages over convention methods, such as 1) it can extract high dimensional eye features from high dimensional image data. 2) It can learn a highly nonlinear function to estimates the gaze. In conventional methods, accuracy drop is observed due to variations like head movement, different illumination conditions, but deep learning-based techniques do not get very much affected by these variations. Also, improvement in performance was observed for cross-subject gaze estimation, which makes it more compatible with real-world application.
Rapid improvement in computing, low-cost hardware, and fast video processing brought eye-tracking products closer to end-users with applications in various domains like web marketing, virtual reality, healthcare, product packaging design, gaming. Various user platforms used eye gaze information in different fields like desktop and mobile-based platform uses eye gaze for computer control and communication, text entry, and gaze-based passwords. Handheld mobile devices like tablets and smartphones use real-time eye tracking information for locking/unlocking phones and different visual interactions. Head-mounted real-time eye tracking set up with multiple external cameras is used extensively to track user attention, cognitive studies, and virtual and augmented reality applications. Real time eye tracking is also used in automated systems like to find driver's and pilot's attention levels. Remote real time eye tracking is used to activate the control function of TV panels. With different use cases, variability in eye movement, external environment, and individual biological aspects pose challenges in achieving consistency in performance from gaze estimation methods. Hence the aim of this work is to provide insights into current gaze estimation research and its accuracy, performance in the real world. This literature presents a detailed overview and analysis that includes algorithm, system set up, user conditions, performance, and evaluation of various methods discussed in multiple works. This work aims to highlight a realistic overview of this field and identify factors that affect the accuracy of real-time eye tracking when applied to practical situations. Specifically, this work will focus on the following: The thesis intends to take on the open challenges in the research field of gaze tracking to develop a system that is both practical and robust, and that can be utilised to construct a retention model for the advertising domain.

The thesis' remaining sections are structured as follows:
Section 2.1 of this chapter explains the foundations of gaze tracking and the motivation for improving the existing gaze tracking techniques. **Section 2.2** covers the fundamentals of eye gaze tracking. It is further classified as follows:

1. The types of eye movement in **Section 2.2.1**.
2. The main setup and method for estimating eye gaze in **Section 2.2.2**.
3. The estimation of gaze tracking accuracy, followed by the conclusion in **Section 2.2.3**.

The user may comprehend the data preparation techniques by studying **Section 3**. It consists of:

1. Data collection methods in **Section 3.1**
2. Data calibration in **Section 3.2**
3. Data preparation in **Section 3.3**
4. Application of lighting normalisation to the data in **Section 3.4**.

Implementation and experimenting with gaze architecture are the main topics of **Section 4**. Additionally, it clarifies the experimental process training and evaluation procedures and how the architecture functions. Finally, it describes how the suggested solution's execution and calibration occur. **Section 5** discusses a more critical issue that an eCommerce company could experience and how gaze tracking might assist them in improving their marketability. It also describes how the suggested solution functions and its effects on the existing businesses.
Finally, **Section 6** will summarise the work done and highlight potential future developments that could improve the suggested approach.

2 Eye Gaze Tracking: Review

2.1 Introduction

This chapter will provide a systematic discussion on several topics related to this field of research. In this literature work, we have provided a comprehensive review of appearance-based gaze estimation using deep learning algorithms. We have discussed it from four perspectives: 1) deep feature extraction, 2) deep neural network architecture design, 3) personal calibration, and 4) device and platform. In deep feature extraction, we divide raw appearance images into eye images, face images, and videos. We have discussed algorithms used for effective feature extraction in deep neural network architecture. We have also discussed CNN models for different supervision methodologies like supervised, semi-supervised and unsupervised gaze estimation methods. We also reviewed different architectures like multi-task CNNs and recurrent CNNs. In personal calibration, we discussed how personal calibration could further improve the accuracy of CNN models. From a device perspective, we reviewed hardware setups like RGB cameras, IR cameras, and depth cameras, and different platforms, like a computer, mobile devices, and head-mount devices. Section 2.2 provides a brief description of Eye Gaze Tracking Fundamentals. Finally, Section 2.3 sheds light on different eye gaze methodologies.

2.2 GAZE TRACKING FUNDAMENTALS

2.2.1 Types of eye movements

To collect information about the user's intent and cognitive behaviour, several types of eye movement are studied[3], these are like:

- Fixations: It refers to the stationary period between eye movement Fixation related measurement variables include total fixation duration, mean fixation duration, number of areas fixated, fixation spatial density, fixation sequences, and fixation rate.
- Saccades: These are involuntary and rapid eye movements between fixation points. Parameter for Saccade measurement includes saccade number, amplitude, and fixation-saccade ratio
- Scanpath: It refers to a number of short fixations and saccades before reaching the final target on the screen.
- Gaze duration: It is the sum of all fixations in a particular area and the proportion of time spent in an area of interest before eyes leave that area of interest
- Pupil size and blink: These measures are used to examine the cognitive workload of a user.

Table 1 presents the significance of different eye movements and applications.

Table 1: Characteristics of different eye movements

Eye movement	Functionality/Significance	Applications in Human Computer interaction
Fixation	Acquiring information, Cognitive processing, attention	Browsing information, reading, scene perception
Saccades	Moving between targets	Visual search
Scanpath	Path traced by user's eye	Assessing user behavior
Gaze duration	Cognitive processing, conveying intent	Item selection, text/number entry
Blink	Indicates behavioral states, stress	Eye liveliness detection, activate command
Pupil size change	Cognitive effort, representing micro emotions	Assess cognitive workload, user fatigue, command

2.2.2 Basic method and setup used for eye gaze estimation:

Video based eye tracking mainly requires one or more digital cameras, near-infra-red (NIR) LEDs, and a computer.

Commonly used steps in eye gaze tracking include methods like user calibration, obtaining video frames of the face and eye regions, detecting eyes, and mapping gaze coordinates on the screen. The most commonly used method is Pupil Center Corneal Reflection or PCCR method. In this method, NIR LEDs are used to produce glints on the eye cornea surface, and then images/videos of the eye region are captured[4]. Gaze is then computed from the relative movement between the pupil center and glint positions. External NIR illumination is also used sometimes for better contrast and to avoid variations produced by natural light. Different gaze tracking methods are discussed in section 2.3. The user interface for gaze tracking can be active or passive, single or multimodal. In an active interface, the user's gaze information is used as an input modality and to activate a function. In a passive interface, eye gaze data is consolidated to predict user interest or attention. In a single modal gaze tracking interfaces, only user's gaze is used as an input variable. In contrast, gaze input is combined with a mouse, keyboard, touch, or blink inputs for command in a multimodal interface.

2.2.3 Estimation of gaze tracking accuracy

In the literature, gaze tracking accuracy measures are reported in different ways, for e.g. angular accuracy in degrees [5],[6],[7],[8], distance accuracy in cm/mm [9], [10], [11, 12, 13, 14, 15, 16], distance in pixels and gaze estimation accuracy in percentages[17, 18, 19].

Some common gaze estimation accuracy is discussed below:

- Gaze point coordinates in Pixels:

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

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

  Where $X_{left}, X_{right}, Y_{left}, Y_{right}$ are the measured X and Y coordinate of the left and right eye's point of gaze (POG)

- Monitor screen pixel size (μ):

$$\mu = \frac{dim_m}{dim_p} \tag{3}$$

  where dim_m is diagonal size of screen in mm and dim_p is diagonal size of screen in pixels as shown below:

$$dim_p = \sqrt{width_p^2 + height_p^2} \tag{4}$$

  Where $width_p$ and $height_p$ is width and height of screen in pixels.

- On-screen distance (OSD): It is the distance between the origin of gaze coordinate system and a specific gaze point. It is calculated by the formula mentioned at the bottom of this page. Where (X_pixels, Y_pixels) = origin of gaze coordinate system and $offset$ = distance between sensor of eye tracker and lower edge of display screen.

- Gaze angle relative to eye: Gaze angle at any point on the screen relative to user's eye can be estimated by using below formula:

$$gaze\_angle(\theta) = \tan^{-1}(\frac{OSD}{Z}) \tag{5}$$

- Distance between eyes and gaze point on the screen: It is given by:

$$ESTGP(mm) == \sqrt{((Gaze\_X)^2 + (Gaze\_Y)^2 + Z^2)} \tag{6}$$

- Pixel Accuracy: Shift between actual gaze coordinates (GT_x, GT_y), and estimated coordinates $(Gaze_X, Gaze_Y)$ can be calculated as:

$$pixel\_shift(pixels) = \sqrt{((GT_x - Gaze\_X)^2 + (GT_y - Gaze\_Y)^2)} \tag{7}$$

- Angular accuracy: Angular accuracy(or prediction error) in degrees can be calculated as:

$$Angular\_accuracy = (\mu * pixel\_shift * cost(mean(\theta))^2)/EstGP \tag{8}$$

- Euclidean distance in cm/mm: Some studies used Euclidean distance to compute error between predicted and actual gaze estimation. Euclidean distance can be calculated by predicted and actual gaze estimation. Euclidean distance can be calculated by:

$$ED = \sqrt{(gt\_x_i - e\_x_i)^2 + (gt\_y_i - e\_y_i)^2} \tag{9}$$

- Where gt_x_i and gt_y_i is ground truth label for each point and et_x_i and et_x_i are estimated gaze coordinates for each points.

Apart from this, some studies simply used distance between predicted gaze and ground truth as error function [16]. Some also uses root mean square error[20] to estimate accuracy.

2.3 Eye Gaze Estimation Methodology

In general, there are 5 types of eye gaze tracking methods: 2D regression based method, 3D model based method, cross ratio based method, appearance based and shape based methods. 2D regression based methods: The method makes use of IR cameras to detect vector of geometric features such as pupil center, glints and directly maps this vector to the POG on the screen using a polynomial transformation function.

Cross Ratio Based Methods: In this method, four or more infrared lights are positioned on the end of the screen (all these are coplanar). A camera is also placed which captures images of user's eyes. The corneal reflection of each light source on the eyes is called glint. From the captured images, we know the glint of each light sources and also the center of pupil. The cross ratio method consider the surface of cornea as a plane and assumes that all the glints are coplanar. A mapping is then performed between the light sources and glints detected by the camera. Once the mapping is done, pupil center can be projected on the screen to estimate POG[21].

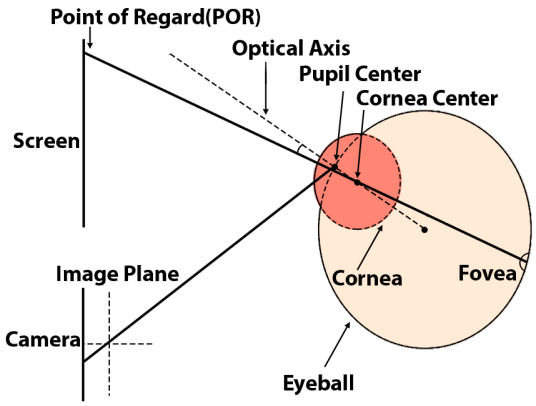

Figure 1: Optical axis joins pupil and corneal lens centres. The visual axis connects the fovea and corneal lens. The point of gaze is the visual axis-screen junction[1]

3D Model Based Methods: This method utilizes geometric 3D model of an eye along with eye features such as pupil center[22], corneal reflection[23] and iris contour[24]. As shown in the figure 1, Optical axis is the line joining the center of pupil and center of corneal lens. The visual axis is a line that runs across the fovea and the centre of the corneal lens. The point of gaze is defined as intersection between visual axis and device screen.
A model based approach for unmodified tablet was presented in [20] to predict gaze direction where 2D ellipse is back projected to 3D to locate eye optical axis. Point of gaze (POG) is then estimated using the intersection of 3D optical axis and display screen. An approach independent of user calibration was presented in [25] which employs multiple camera and multiple point light sources to estimate gaze direction.
Appearance Based Methods: Appearance based methods directly learns mapping from input images to point of gaze. As in this method there are many variability in unconstrained environment, so conventional appearance-based methods cannot handle these variation due to the weak fitting ability. Since convolutional neural networks (CNNs) and other deep learning-based algorithms can handle large datasets, they will be utilised to detect gaze.
The several gaze estimation algorithms presented above have distinct characteristics, merits and demerits. The 2D regression based methods makes use of the features of the human eye and can be incorporated using a few NIR LEDs and a single camera. However, these techniques are very much affected by head movements and often require users to keep their head stationary.
Cross ratio based methods do not need an eye model or hardware calibration and allow free head motion. But distance between the user and screen needs to be kept constant. 3D model based methods allows free head movement but the hardware requirements for implementing 3D methods are high as they need several illumination sources or multiple cameras.
Appearance based methods are non-PCCR methods that directly learns mapping from input images to point of gaze. These methods have low hardware requirements but disadvantage is that their gaze estimation accuracy is slightly less than PCCR based methods. However in recent years, with the evolution of deep learning convolutional neural network

(CNN) architecture, appearance based methods are widely used in gaze estimation. Accuracy of above methods can be increased by increasing number of calibration points. We use following structure to review work in this domain: Feature Extraction, Gaze estimation methodology, Calibration, Devices and Platforms.

2.3.1 Feature Extraction:

Features extraction from various methods (for e.g. appearance based, model based etc.) is a challenging task due to complex eye appearance. Since accuracy of gaze estimation depends largely on quality eye features hence we have discussed various methods to extract features like extracting features from eye images, from face image, from videos. Gaze estimation is highly dependent on eye appearance. Rotation of eyeball can change gaze direction. This dependency makes it possible to use eye images to estimate gaze direction.
Features from single eye or both eyes can be extracted. Zang et al. [26] proposed LetNet CNN network to extract eye features from grey-scale single eye images and merge these features with an estimated head pose. In [27] he further modified his previous work with GazeNet which is a 13-convolutional layer neural network inherited from a 16-layer VGG network to extract the individual features from single eye and to find gaze. Fischer et al. [28] also implemented a two VGG-16 networks to extract individual features from two eye images, and merge these features for regression. Cheng et al. [29] used a 4 stream CNN where two streams are used to extract individual features from eye and remaining two streams are used to extract common features from both eyes. Bao et al. [30] proposed a self-attention mechanism to merge two eye features. They merged the feature maps of two eyes and used a convolution layer to generate the weights of the feature map. Cheng et al. [31] assigned weights to both eye features based on the guidance of facial features . Wood et al. [20] extracted eye features using cascade classifier and a shape based approach.

Several works have aimed to extract subject invariant traits from eye images [5], [32]. Wang et al. proposed an adversarial learning approach to extract the domain/person invariant feature [33]. The eyeball, iris, and pupil are represented pictorially in Park et al.'s[5] work by transforming the original eye photographs. They use an auto encoder to learn the compact representation of gaze, head pose and appearance. Fischer et al. [28] used a GAN to remove eyeglasses from images. In addition to above, unannotated eye images can also be used for gaze representation.
Various studies have considered face images as input to estimate gaze as they contain information about head poses. Head pose contributes plays a crucial role in gaze estimation to overcome Wollaston effect[34].The extracted features contain facial landmark and head pose [34], [28]. Various studies only uses face images as input and implement a CNN to automatically extract deep facial features [8], [14].
Some works filter unnecessary features from face images as they contain unnecessary information. Zhang et al. [8] proposed a spatial weights mechanism to reduce noise and to efficiently extract information about different regions of the face. The spatial weights applied on the feature maps are then fed to a CNN architecture to estimate gaze. Zhang et al. [35]proposed an architecture to extract information dynamically based on the image content from input images.
Some works crop the eye images from the input face images and then feed it to network. Cheng et al. [31] proposed a coarse-to fine gaze estimation method. They first extracted coarse grain features input face image and then fine grained these features to estimate gaze. Palmero et al. [34] combined facial landmark along with face and eye region to detect gaze. Jyoti et al.[9] used facial landmarks to extract geometric features like angle between the pupil centers and facial landmarks of the eyes and tip of the nose as shown in figure 2. Dubey et al. [19] collected images from YouTube videos and then implemented unsupervised learning based method to estimate gaze.

In addition to face and eye images, videos can also be used for feature extraction. From images, we get information about static features only. However temporal information, which can be obtained from videos can be incorporated as an added advantage for better gaze estimation. Recurrent Neural Network (RNN) has been generally used in video processing like long short-term memory (LSTM) [18], [36] as RNN architectures can retain sequence information also. Zhou at el [36] applied many to one bidirectional LSTM to fit the temporal information between frames to predict gaze vector for video sequences.

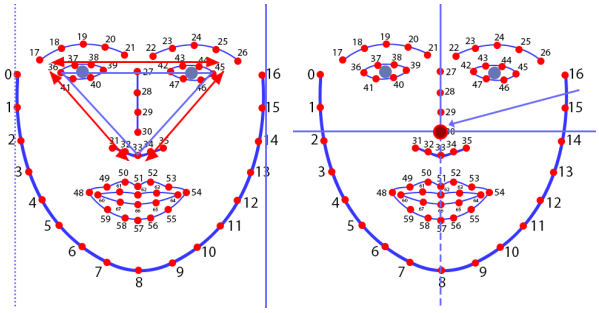

Figure 2: Face Landmarks using dlib 68 Facial Landmarks Detector

2.3.2 Gaze Estimation Methodology:

This section will cover a variety of approaches used in diverse works. First, we have discussed model based methodologies and then for appearance based gaze estimation with a major focus on Convolutional Neural

Network architecture based appearance gaze estimation.

Wood et al. [20] presented a model based approach for unmodified tablet to predict gaze direction. This system does not require any external camera or any infrared illumination. However, distance between the user and display was fixed. In this, first rough eye features from the image are extracted using cascade classifiers and eye centers are then located using a shape based approach. Limbus ellipse fitting is then performed along the eye region of interest (ROI) with robust model-fitting approach. The obtained 2D ellipse is then back projected to 3D to locate the eye optical axes. Point of gaze (POG) is then estimated using the intersection of 3D optical axis and display screen. Shih et al. [25] presented an approach independent of user calibration which employs multiple camera and multiple point light sources to estimate line of sight. Two light sources and two cameras that are not collinear are used to find the 3D locations of pupil and cornea centers. Gaze estimation is then performed by connecting the pupil and cornea center.

For appearance based gaze estimation, convolutional neural network architectures have been widely used as these networks give better performance. Also they require few input parameters for estimating gaze. Supervised models are most widely used method for appearance based gaze estimation. In these models, CNN network is trained using image samples along with ground truth gaze directions. It is basically learning a mapping function from raw images to gaze directions [26], [37], [6] .Many CNN architectures which are used for computer vision tasks can also be implemented for gaze estimation like LeNet [26], Alex Net [8], VGG [27], ResNet18 [18] and ResNet50 [38]. Zhang et al [26]used LeNet network architecture that comprises of one convolutional layer along with max pool layer, a second convolutional layer along with max pool layer and a final fully connected layer with linear regression at the top to predict gaze vectors. Head vector is added to the output of the fully connected layer. Input to the architecture are gray scale images of size 60 x 36 pixels. The number of feature layer for the two convolutional layers is 20 and 50 for the first and second layer respectively with a feature size of 5 x 5 pixels. In the fully connected layer, the number of hidden layers are 500. The output of the network is a 2D pitch and yaw gaze angles. Zhang et al. [8] used a CNN with spatial weights for 2D and 3D gaze estimation. He used Alex Net CNN architecture that consists of five convolutional layers and two fully connected layers. The input image of size 448 x 448 pixels is passed through this CNN architecture to generate feature vector of size 256 x13 x 13. This feature vector is then passed through spatial weights mechanism which consists of a convolutional layer with filter size of 1 x1 followed by a rectified linear unit to generate a weight map which is again multiplied with the feature vector extracted before using element wise multiplication. Depending on the task whether to find 2D or 3D gaze estimation, the output from the element wise multiplication is then fed to the corresponding fully connected layer to find gaze direction. Zhang et al. [27] developed a new architecture called GazeNet. It is based on a 16 layer VGG architecture consisting of 13 convolutional layers along with two fully connected layers and one classification layer. Grey-scale image of resolution 60 x 36 pixels is used as an input. The output of the network are 2D pitch and yaw gaze angles. Apart from these architectures, some well-designed modules also help to improve the estimation accuracy [39], [31]. Chen and Shi [39] developed an architecture based on dilated convolutions where given a kernel of a particular height and width, we insert a spaces(zeroes) between the weights so that kernel covers a larger region than the given height and width. Input to this architecture are face image of size 96 x 96 pixels and two eye images of size 64 x 96 pixels. The face network consists of four stacked convolutional layers followed by max-pooling layers along with two fully connected layers at the top. The two eye networks shares same weights and consists of four convolutional layers along with max-pool layer in the middle , followed by 1 x 1 convolutional layer and one fully connected layer. Rectified Linear Unit (ReLU) activation function is applied in all the layers. The outputs from the different networks are combined together and fed to another fully connected layer to estimate gaze.

We need large scaled labeled dataset such as MPIIGaze [27] and Gaze Capture [14] to supervise CNN during training. But collection of such a huge amount of data is very difficult and time consuming hence some researcher used synthesized labeled photo-realistic image [40]. These techniques first build an eye-region models and then render new images from these models. Wood et al. [41] proposed to synthesize the close-up eye images for head poses, gaze directions and illuminations in large scale to develop a robust gaze estimation algorithm. To make synthesized images more close to the real ones. Fisher et al. [28] also implemented a GAN based image inpainting method to remove eye tracking glasses.

Another type of CNN architecture is semi supervised CNN which requires both labeled as well as unlabeled data to optimize the CNN Network. Cheng et al. proposed a self-supervised asymmetry regression network for gaze estimation [29]. It contain two network one is regression network to estimate the gaze from two eye images. It also provides ground truth which can be used to train the other network. Second network is an evaluation network to assess the reliability of two eyes. The proposed network takes two eye images and head pose vector as input. The first network is a four streamed convolutional network consisting of six convolutional layers along with three max pooling layers and a fully connected layer at the end. The second network is a two stream convolutional neural network consisting of six

convolutional layers with three max pooling layers along with three fully connected layers at the end. The first network finds asymmetric regression of the two eyes and the second network works on improving gaze estimation accuracy. During training, both networks train each other simultaneously like result of regression network is used to supervise the evaluation network and accuracy of evaluation network is used to optimize the learning rate of regression network. He et al. [11] used a person-specific user embedding mechanism to estimate gaze with very few calibration points. For estimate the gaze, they concatenated the user embedding with appearance features .They have developed a teacher -student networking system in which during training teacher network optimize user embedding and student network learn from teacher network. The input to the network are eye landmark features, both eye images and unique id. The network consists of three convolutional layers followed by average pooling layers along with five fully connected layers.

Unsupervised CNN network need only unlabelled image data for training purpose but it takes more time to optimize the CNN network. Dubey et al. [19] collect unlabelled facial image from web for gaze representation learning. They approximately annotated the gaze region based on the detected landmarks. Even though these approaches can learn the gaze representation, but then also few labelled samples are required to fine-tune the final gaze estimator. He proposed a novel architecture "Ize-Net". The network takes entire face image of size 128 x128 x3 as a input. Five convolutional layers are used, then batch normalisation and maximum pooling are applied. The output is then fed to fully connected layers of size 1024 and 512 with a 'SoftMax' activation at the end to estimate gaze.

For improving model generalization we can use multi-task CNN. It contains multiple tasks that provide related domain information which can improve robustness of model [7], [42]. Lian et al. proposed a multi-task CNN based learning network to estimate point of gaze by using depth images [42]. They first extracted eyeball features from two single-eye images, head pose features from RGB and depth images with the help of GAN. Then depth values of eye region and original eye coordinates are used to encode 3D eye position. All the features are then concatenated and fed in to a network for gaze estimation. They also collected a large scale RGBD dataset for performance evaluation.

Yu et al. introduced a constrained landmark gaze model (CLGM) for modelling eye landmark locations and gaze directions [7]. They first estimated the coefficients of a joint CLGM landmarks-gaze model as well as the scale and translation parameters that define the eye region. Gaze is then estimated using head poses and CGLM coefficients. Deng et al. [43] used two individual CNNs to find head pose and eye ball movement. A gaze transform layer will then combine the results from the above two CNNs for gaze prediction.

In recent years, Recurrent neural networks is widely used to estimate the gaze in videos as it has been observed that recurrent neural networks have shown good capability in handling the sequential data frame [18], [34], [36]. Since human gaze is continuous, Kellnhofer et al. [18] proposed a video based gaze tracking model implemented on bidirectional Long Short-Term Memory capsules (LSTM) where outputs are dependent on both past and future values. Multiple frames of input are fed to a backbone model first to extract high level features. The extracted features are then fed to bidirectional LSTMs with two layers along with a fully connected layer to get gaze prediction.

Palmero et al. [34] also implemented a many-to-one recurrent network. The recurrent network extracts sequential information to predict 2D gaze angles. The network is divided in to 3 modules: individual, fusion and temporal module. Input to the individual module are full face images (224 x 224 pixels), eye region (120 x 48 pixels) and facial landmarks. Individual module consists of 13 convolutional layers along with 5 max pooling layers and 1 fully connected layer with a Rectified Linear Unit (ReLU) activation function. The fusion module consists of two fully connected layers along with ReLU activations and two dropout layers as a regularization. All the models are trained using ADAM optimizer with an initial learning rate of 0.0001, batch size of 64 frames. Average Euclidean distance is used as a loss function between the predicted and actual ground truth. The temporal module consists of many-to-one recurrent network and extracts sequential information to predict 2D gaze angles. Zhou et al. [36] first extracted features from face and eye images. The extracted features are fed to bidirectional LSTM to secure temporal information between frames for estimating gaze vectors for videos. It consists of two modules, one is static and other is temporal modules. The static module consists of a two branch convolutional neural network and one fully connected layer. The input to the static module consists of normalized face images and two eye images, both of resolution 224 x224. One branch of convolutional neural network extracts features from the face and other branch from the eye images. The fully connected layer combines these results from two branches which are then fed to many to one bi-directional LSTM. A linear regression is then used to predict gaze in the last time stamp.

2.3.3 Calibration

The eye parameters usually required in gaze estimation are pupil center, center of corneal lens, the optical and the visual axes. As shown in figure 1, the back of the eyeball is called retina and a place on the macula of the retina where sharpest

image is formed is called as fovea. In general, Calibration can be performed via active and passive method. In active method, Calibration is performed by showing user a fixed number of points on the screen (see figure 3). Each user is asked to gaze at these points for a certain period. Offset is then calculated between real gaze and estimated gaze. Whereas in passive method, user is asked to do regular device usage and as they are using their devices, there are certain things which require users to fixate on certain locations. Using this passive information, calibration is then performed to improve accuracy.

The common approach adopted in many works is to fine tune the model while testing it on unseen data[14], [44]. Krafka et al. [14] implemented an SVR in place of a fully connected layer in the last and fine-tuned it to predict the gaze location. With calibration, there error got reduced by 0.40 cm. Zhang et al. [44] performed both implicit and explicit calibration. The entire CNN network was divided in to three parts: the encoder, the feature extractor, and the decoder. Encoder and decoder was fine tuned in each of the five devices used in the study (mobile devices, tablet, laptop, desktop, smart tv). In external calibration, participants were asked to fixate on a circle and perform a click when circle gets converted into a dot. While in implicit calibration, face videos, timestamp and location of interaction events were recorded to collect ground truth. In another work [15], firstly user was asked to fixate on the circle until it become a dot. The samples obtained are then used to get a third-order polynomial mapping function between the estimated and ground-truth 2D gaze locations.

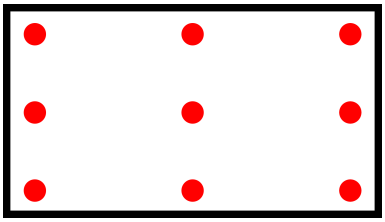

Figure 3: Active calibration method

He et al. [11] proposed a supervised calibration method with embedding based few shot learning using only 2-5 calibration points instead of more than 13 points often used in most of the works. They also proposed unsupervised personalization method to improve accuracy based on teacher-student framework using few unlabeled images.
Park et al. [32] proposed another algorithm for person specific gaze with very few calibration samples (<9). They used a Meta learning based calibration methodology. Using meta-learning, an adaptable gaze estimation network was trained to implement person specific gaze estimation network. Liu et al. [45] proposed a network such that if a subject specific calibration images are given, then the network can predict gaze of any novel sample.

Most of the available calibration methods are supervised and need labeled samples .But to collect such a large amount of data is very cumbersome and time taking so an alternate way is to collect calibration sample in a user unaware manner [10]. Chang et al. [10] introduced a framework SalGaze. It makes use of saliency information such that gaze estimation algorithm will automatically be adapted for new user without any explicit calibration.

In many works, calibration is not performed [20], [15]. Even though calibration was not performed, they are able to estimate gaze with very less error. With the advent of better image capturing devices, need of calibration is decreasing day by day and continuous research in going in this area.

3 Data Prepration and Preprocessing

3.1 Data Collection

To perform data processing we first collect data samples via custom script designed in python. The objective was to collect the images of the users while using the application as they would be doing activities of daily life while using a laptop. Users were shown the red dots on the screen as shown in the figure 3.

3.2 Calibration Window

The users were asked to stare the dots one by one. The dots were shown over the screen one after the other in fashion. The dots were shown on 1 seconds and other dots followed has the time interval of 0.2 sec. The image were captured for user at 5 fps that is 5 images per second. The experiment was performed 2 times for the user at different interval.

3.3 Data Preprocessing

To train model we model the data normalised of same dimension and resolution thus every image was resized to 512 X 512 pixel. Further we pass images to mediapipe face tracker with confidence interval of 0.5 to see if the model was able to detect face and puple or not. If the face was not recognised the image was discarded. Further using the mediapipe landmark detection the eyes, nose and jaw points were extracted and bounding box was created for both eyes and face.

Further image was cropped and saved accordingly for every subject. Reference figure shows the folder structure of the data were the subject has left eye followed by coordinate and right eye followed by the coordinate, face followed by the coordinate. Figure 4 shows folder structure of the data.

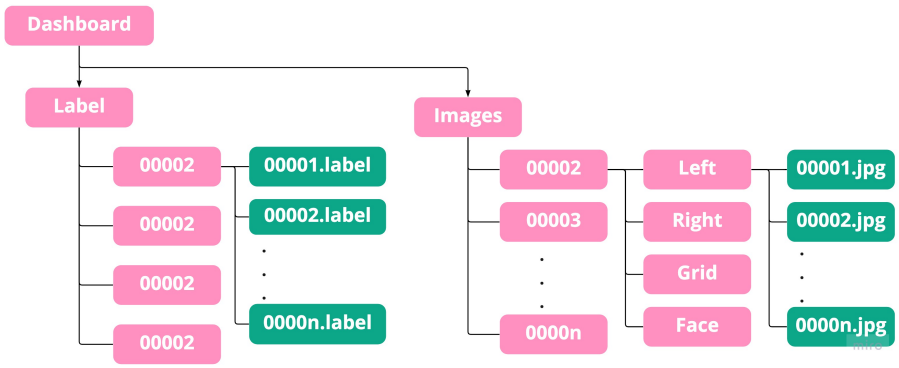

Figure 4: Folder Structure

3.4 Lighting Normalization

To normalise the data we first need to perform lighting normalization of the data. To normalise the light we choose any image as reference image and remove the specaluratiy from the image. To remove the specularity from the image we consider image I and convert it to grey scale. The specular image can modeled as:

$$I = \alpha + (1 - \alpha) * I_c \tag{10}$$

Where I is the reference image and α contains the specularity map and Ic is the clean image. Firstly to extract clean image, specularity map needs to be obtained. To do this, firstly the image was smoothed and the pixel values higher than 90th percentile was discarded in the RGB channel. Further using whole filing method the images were filled with values of neighbouring pixel . The α was calculated by

$$\alpha = (I - I_c)/(1 - I_c) \tag{11}$$

Final clean image is obtained by

$$I_c = (I - \alpha)/(1 - \alpha) \tag{12}$$

Since our hypothesis was that the image has been captured asking the user to sit in fixed position we perform the lighting normalization under the same assumption.

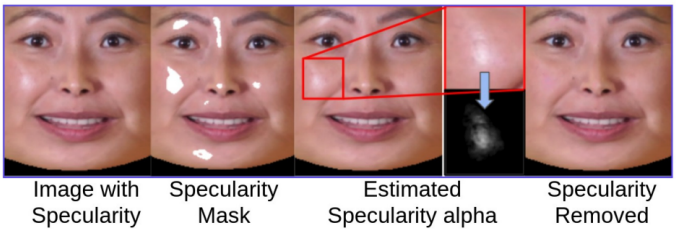

Figure 5: Smoothing the input image and eliminating pixels with RGB values above 90 removed specularity. Images were filled with neighbouring pixel values using whole filing[2]

4 Implementation and Discussion

4.1 Experiment

The proposed solution uses appearance based gaze estimation, convolutional neural network architecture. These networks give better performance compared to traditional approaches. The model used in this experiment comprises three overall networks concatenated at the end. Following are the three networks used for creating the model:

4.1.1 Left and Right Eye

The architecture for both left and right eyes is the same. The convolutional network takes in the image as an input with the size of 224x224 and three channels and passes the image through a convolutional layer of kernel size 11, stride 4, padding 0. The output converts three-channel input to 12-channel output, equalling 54x54. Further, Relu is used to introduce non-linearity.
Next, the output from the previous layer is used as input for this layer. Therefore an image of size 54x54 with 12 channels is passed through Maxpool2D with kernel size three and stride as 2. This results in a size reduction to half; therefore, the output is of size 27 x 27, but the number of channels remains unchanged.
The output is then passed to another convolutional layer where kernel size is 5, stride 1, padding 2. The result has 32 channels, and the size remains unchanged. Relu is again used to introduce non-linearity.
Further, max-pooling with kernel size three and stride as two is applied on an image that reduces the size to 13x13 and channels remain the same. LRN2D is then used to normalise the output. Another two layers comprise of convolution layer with relu as an activation function in between with kernel size, stride, padding, input size and channel as 3, 1, 1, 13x13 and 32 as a first input and 1,1,0,13x13 and 48 as second input and dimensions equal to 13x13 with eight channel as the final output.
In the end, we have features for both eyes that are further concatenated so that the model works for both eyes. Additionally, the output is connected to a fully connected layer where input is taken as 13x13x8x2 and is connected to 16 hidden neuron layers.

4.1.2 Face

A fully linked layer is added on top of the convolutional architecture that was used to model the face. This is done in a manner that is analogous to the construction of an individual eye. In this particular scenario, the hidden layers of 13x13x8 convolutional computations produced by 16 neurons are connected, resulting in eight outputs.

4.1.3 Grid

Flattening the 25x25 black image results in its connection with 32 neutrons. Those neutrons, in turn, are associated with 16 neurons, each of which is has relu as an activation function.

4.1.4 Fully Connected Layer

So the output from all the above three(Eyes, Face, Grid) are further connected with 16 hidden neurons; each connection is processed with the relu activation function. This provides a binary output with a relu activation function. The figure 6 depicts the complete architecture.

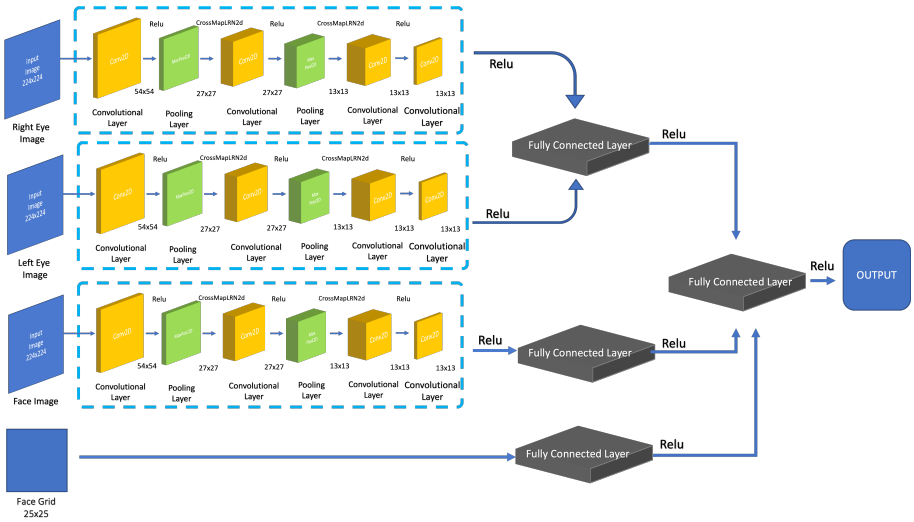

Figure 6: Gaze Tracking Training Architecture

4.2 Training

The pre-processed images were trained over our constructed architecture for 100 epochs with batch size 400. It is trained over 8 GB Nvidia GeForce RTX2070 Super GPU. The loss function used for our training is MSE loss(mean square error loss). The optimiser used in training is stochastic gradient descent, and hyper-parameters used are momentum and weight decay. Momentum was initialised to 0.999 with a weight decay of 0.0001, and the learning rate used was 0.00001. Simple backpropagation was used to update weights, and at every epoc, weights and loss were stored in our model file for analysis.

$$MSE = 1/n \sum_{n} (y_i - \tilde{y}_i)^2 \tag{13}$$

4.3 Evaluation and Execution

4.3.1 Evaluation

As mentioned before, the loss function used in the experiment is MSE and the loss at every epoc is shown in figure 7. The best loss is 6.23080015.

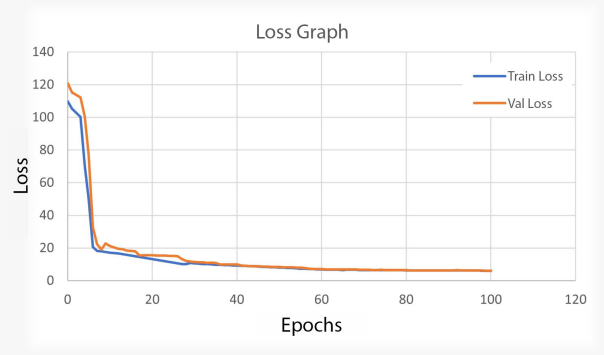

Figure 7: Loss Graph

4.3.2 Execution

The final model was validated by performing tests on a test version of our very own web application. When the user is browsing the website, they will get a pop-up asking for their permission to see the live camera feed. Once the user gets access to the camera, they will be directed to a website where they can browse for and rate products or to a calibration page with nine dots that must be clicked on five times to calibrate the programme in real time. Both of these options are available once the user has been granted access to the camera.

4.3.3 Calibration Process

The calibration of the gaze tracking takes place in four stages as shown in figure 8, which are as follows:

1. The web browser will request permission to access the camera. A window that asks you to begin the calibration process appears.
2. After that, a popup will appear with instructions on how to calibrate the device correctly.
3. To complete each task, you will need to click on the dots that appear on the screen five times.
4. At long last, a popup will appear that shows how accurate the calibration was.

Following these steps will result in more accurate gaze tracking. A non calibrated implementation doesn't take in the calibration data but is less accurate in real time usage.

5 Customer Retention-Application

Every day, new companies in the industry are attempting to establish their businesses online in response to the steadily increasing demand for eCommerce. However, most eCommerce businesses fail because their owners lack knowledge of the best marketing strategy for their enterprise. Although some firms use digital marketing to increase the number of customers on the platform, most cannot draw visitors to their website. This results from the lack of information regarding the customer or the products they look up online. Various methods are used to collect client information, such as their phone number, email address, or the products they

searched for online. However, most of these attempts fail to address one of the most critical aspects of the user journey, i.e. the kinds of products that are affecting them, without soliciting any feedback or surveys. The goal of the software/web application is to capture the eyesight of the customer and identify what the user was interested in or how much time the user focused on a specific region of the website. These can range from a basic search filter to the type of search, time spent on a specific product page, longer gaze at specific colour/type of product, etc. The implemented remedy functions in three steps:

1. Capture the gaze of the user throughout the website.
2. Real-time data collection of the user's focus points and time.
3. Next time, make better search results and adverts that the user could find interesting using the data acquired.

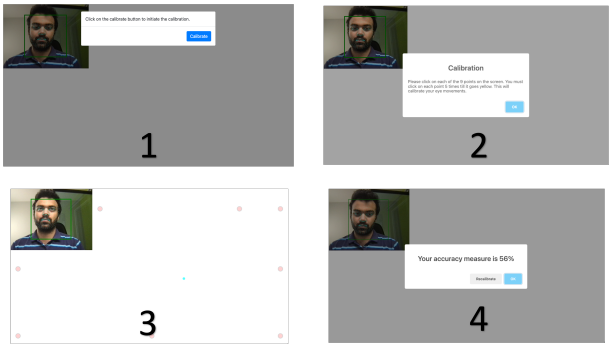

Figure 8: Calibration Screen

5.1 Output

The approach above will produce better product recommendations that the user is interested in. This information, along with the search information and previous purchases, can be used to build a recommendation system that learns about the user's preferences and suggests appropriate products, automatically applies filters or produces small insights like the user's preferred colour, the length of the product description, the pricing they are interested in, and much more. One can develop a competitive eCommerce that can compete with the established eCommerce giants by catching the user's attention.

6 Conclusion and Future Work

Over the last few decades, eye gaze estimate has garnered significant interest from various industries, academic institutions, and other fields. E-Commerce has also made its market with an ever-growing customer base. This paper discussed a detailed study of gaze estimation methods to highlight diversity in various aspects such as gaze estimation basics, feature extraction, architecture implemented to estimate gaze, calibration, datasets, and performance measures implemented in multiple works. It also highlights how these implementations can benefit the eCommerce industry by enhancing customer experience.
The literature highlights a unique implementation of appearance-based gaze estimation, a convolutional neural network that performs better than traditional approaches. The overall experiment performed was successful and the best validation loss achieved was 6.23080015.
Lack of homogeneity is observed in performance evaluation among several works. Some works estimated accuracy in percentage; others measured accuracy in terms of distance or degrees. This variation makes inter-comparisons between different implementations improbable. Even though CNN-based deep learning architectures are very effective in estimating gaze, these methods also have some limitations that can become the basis of future works:

1. One can implement a more complex model to reduce the loss even more.
2. Improving the GPU will result in less loss due to parallel processing. Also, the computation time will decrease by 2-3 folds.
3. Future research can focus on developing hardware-friendly, computationally inexpensive architectures that do not require external GPUs or multi-core CPUs for smooth implementation. A lighter implementation might be a way to go forward in real-time applications.

References

[1] Kang Wang, Shen Wang, and Qiang Ji. Deep eye fixation map learning for calibration-free eye gaze tracking. In *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*, pages 47–55, 2016.

[2] Avisek Lahiri, Vivek Kwatra, Christian Frueh, John Lewis, and Chris Bregler. Lipsync3d: Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2755–2764, 2021.

[3] Robert JK Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The mind's eye*, pages 573–605. Elsevier, 2003.

[4] Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006.

[5] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–738, 2018.

[6] Joseph Lemley, Anuradha Kar, Alexandru Drimbarean, and Peter Corcoran. Convolutional neural network implementation for eye-gaze estimation on low-quality consumer imaging systems. *IEEE Transactions on Consumer Electronics*, 65(2):179–187, 2019.

[7] Yu Yu, Gang Liu, and Jean-Marc Odobez. Deep multitask gaze estimation with a constrained landmark-gaze model. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[8] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 51–60, 2017.

[9] Shreyank Jyoti and Abhinav Dhall. Automatic eye gaze estimation using geometric & texture-based networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2474–2479. IEEE, 2018.

[10] Zhuoqing Chang, J Matias Di Martino, Qiang Qiu, Steven Espinosa, and Guillermo Sapiro. Salgaze: Personalizing gaze estimation using visual saliency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[11] Junfeng He, Khoi Pham, Nachiappan Valliappan, Pingmei Xu, Chase Roberts, Dmitry Lagun, and Vidhya Navalpakkam. On-device few-shot personalization for real-time gaze estimation. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[12] Tianchu Guo, Yongchao Liu, Hui Zhang, Xiabing Liu, Youngjun Kwak, Byung In Yoo, Jae-Joon Han, and Changkyu Choi. A generalized and robust method towards practical gaze estimation on smart phone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[13] Matthew Kim, Owen Wang, and Natalie Ng. Convolutional neural network architectures for gaze estimation on mobile devices. *Standford, CA: Standford University.[Google Scholar]*, 2016.

[14] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[15] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Evaluation of appearance-based methods and implications for gaze-based applications. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13, 2019.

[16] Liu Jigang, Bu Sung Lee Francis, and Deepu Rajan. Free-head appearance-based eye gaze estimation on mobile devices. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 232–237. IEEE, 2019.

[17] Anjith George and Aurobinda Routray. Real-time eye gaze direction classification using convolutional neural network. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2016.

[18] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6912–6921, 2019.

[19] Neeru Dubey, Shreya Ghosh, and Abhinav Dhall. Unsupervised learning of eye gaze representation from the web. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[20] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the symposium on eye tracking research and applications*, pages 207–210, 2014.

[21] Jia-Bin Huang, Qin Cai, Zicheng Liu, Narendra Ahuja, and Zhengyou Zhang. Towards accurate and robust cross-ratio based gaze trackers through learning from simulation. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 75–82, 2014.

[22] Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2):802–815, 2011.

[23] Zhiwei Zhu and Qiang Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE Transactions on biomedical engineering*, 54(12):2246–2260, 2007.

[24] Kenneth Alberto Funes Mora and Jean-Marc Odobez. Geometric generative gaze estimation (g3e) for remote rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1773–1780, 2014.

[25] Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. A calibration-free gaze tracking technique. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 201–204. IEEE, 2000.

[26] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.

[27] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.

[28] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European conference on computer vision (ECCV)*, pages 334–352, 2018.

[29] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 100–115, 2018.

[30] Yiwei Bao, Yihua Cheng, Yunfei Liu, and Feng Lu. Adaptive feature fusion network for gaze tracking in mobile tablets. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9936–9943. IEEE, 2021.

[31] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10623–10630, 2020.

[32] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9368–9377, 2019.

[33] Kang Wang, Rui Zhao, Hui Su, and Qiang Ji. Generalizing eye tracking with bayesian adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11907–11916, 2019.

[34] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *arXiv preprint arXiv:1805.03064*, 2018.

[35] Xucong Zhang, Yusuke Sugano, Andreas Bulling, and Otmar Hilliges. Learning-based region selection for end-to-end gaze estimation. In *BMVC*, 2020.

[36] Xiaolong Zhou, Jianing Lin, Jiaqi Jiang, and Shengyong Chen. Learning a 3d gaze estimator with improved itracker combined with bidirectional lstm. In *2019 IEEE international conference on Multimedia and expo (ICME)*, pages 850–855. IEEE, 2019.

[37] Bhanuka Mahanama, Yasith Jayawardana, and Sampath Jayarathna. Gaze-net: Appearance-based gaze estimation using capsule networks. In *Proceedings of the 11th augmented human international conference*, pages 1–4, 2020.

[38] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[39] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018.

[40] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1821–1828, 2014.

[41] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3756–3764, 2015.

[42] Dongze Lian, Ziheng Zhang, Weixin Luo, Lina Hu, Minye Wu, Zechao Li, Jingyi Yu, and Shenghua Gao. Rgbd based gaze estimation via multi-task cnn. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 2488–2495, 2019.

[43] Wangjiang Zhu and Haoping Deng. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3143–3152, 2017.

[44] Xucong Zhang, Michael Xuelin Huang, Yusuke Sugano, and Andreas Bulling. Training person-specific gaze estimators from user interactions with multiple devices. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[45] Gang Liu, Yu Yu, Kenneth A Funes Mora, and Jean-Marc Odobez. A differential approach for gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):1092–1099, 2019.

GAZE TRACKING BACKGROUND: SUPPLEMENTARY

Dhruv Rohatgi
School of Computing
Newcastle University
Newcastle Upon Tyne
d.rohatgi2@newcastle.ac.uk.com

Supervisor: Prof. Boguslaw Obara
School of Computing and Biosciences Institute
Newcastle University
Newcastle Upon Tyne
Boguslaw.Obara@newcastle.ac.uk

ABSTRACT

Knowing the direction in which a person is looking can shed light on their intentions and provide clues as to what they are interested in, making gaze estimate an invaluable resource. Many computer vision problems, such as gaze estimation, have benefited greatly from the recent developments in deep learning algorithms. The development of deep learning algorithms for gaze estimation, however, is hindered by a lack of standard practises. Lack of real-world application is also a factor discouraging research in this area. This article provides a thorough analysis of existing methods for gaze estimation. The results of this research will hopefully aid in the creation of more accurate gaze estimate models and equip researchers with the tools they need to do their jobs better. For those interested in training their own deep learning models, this article lists a number of available pre-trained models, network designs, and open-source datasets.

In this work, we provide a detailed description of all publicly available gaze estimation datasets and a brief summary of gaze estimation algorithms in tabular form, allowing for direct comparison of their respective performance. Besides pointing the way toward future studies on gaze estimation, this study can be used as a reference when creating deep learning-based gaze estimation systems. Finally, a comprehensive overview of the literature around the use of eye gaze estimation in a variety of settings is provided. These settings range from human-computer interaction and psychology to marketing and computer vision.

***K*eywords** Gaze Tracking · Computer Vision · Dataset · Machine Learning

Contents

1 Introduction **4**

2 Background Study **4**

2.1 Artificial Intelligence 4

2.1.1 Importance of AI 5

2.2 Machine Learning 6

2.2.1 Importance of Machine Learning 6

2.3 Deep Learning 8

2.3.1 Deep Learning Methods 10

2.4 Convolutional Neural Network 11

2.4.1 Convolutional Layer 11

2.4.2 Pooling Layer 12

2.4.3 Fully Connected Layer 13

2.5 Segmentation 13

2.5.1 Encode-Decider Models 13

2.6 Residual Networks-ResNet 14

3 Eye Gaze Tracking: Review **15**

3.1 Datasets 15

3.2 Results and Discussions 16

3.3 Device and Platforms 17

4 Conclusion **19**

List of Figures

1 Pictorial Overview Of AI System . . . 5
2 Abilities Of AI . . . 5
3 Types Of Machine Learning Methods . . . 7
4 Examples of under-fitting, appropriate and over-fitting . . . 8
5 Difference between deep learning and machine learning . . . 10
6 Basic CNN Architecture for classification . . . 12
7 Example of Average Pooling . . . 12
8 Example of Max Pooling . . . 13
9 Semantic Segmentation Vs Instance Segmentation . . . 14
10 UNet Architecture . . . 15

List of Tables

1 Summary of Gaze datasets allowing continuous head poses and gaze directions. . . . 12
2 List of image segmentation models . . . 14
3 Summary of Gaze datasets allowing continuous head poses and gaze directions. . . . 16
4 Summary of Gaze Estimation Methods applicable to Desktops . . . 18

1 Introduction

Eye contact is widely recognised as one of the earliest and most fundamental examples of passive communication. In the past few decades, advancements in eye gaze tracking technology have led to the development of effective gaze estimation approaches for human-computer interaction. These techniques have been made possible by the convergence of eye gaze monitoring and computer graphics. The first step involved placing skin electrodes around the eyes for the purpose of gaze prediction. Eye trackers that are worn on the head emerged on the market after years of development and study. After the year 2000, as a result of the rapid advancement in the processing speed of computers, several methods for gaze estimation have been proposed. These methods include the 2D regression model-based method, the 3D eye model-based method, and the appearance-based method. The primary focus of these methods is on increasing accuracy and reducing the number of constraints placed on the user. A transformation function is used in a method that is based on a 2D model, and this function transfers the feature vector directly to the point of gaze (POG). In order to determine where a person is looking, the method that uses a 3D model builds a geometric model of the eyes. Either a two-dimensional or a three-dimensional model of regression requires a dedicated complicated apparatus, such as infrared cameras. Appearance-based methods are non-PCCR approaches that directly learn the mapping from input images to the point of gaze. These methods are also known as "lookalike models." With the recent growth of deep learning algorithms, designs for gaze estimation that are based on convolutional neural networks are quickly gaining a lot of popularity. These deep learning models are able to directly map input features to gaze direction without the need for any external calibration or with very few steps of calibration compared to other current approaches. The appearance-based gaze estimate method does not require a complicated setup; all that is required is a camera to capture the appearance of human eyes. We present an exhaustive background study of machine learning and its fundamentals, as well as a comprehensive assessment of appearance-based gaze estimation utilising deep learning algorithms, in this body of work that we have compiled from the available literature.

2 Background Study

2.1 Artificial Intelligence

The phrase "artificial intelligence" is made up of two words: "artificial" and "intelligence". "artificial" refers to something natural but was created by humans, and "intelligence" refers to the capacity to solve problems and learn from experience. The term "intelligent" can be applied to computers if they can solve problems that occur in the real world by gaining knowledge from their past errors and growing on their own. This is because such machines are typically computerised. Therefore, the pretence that machines, especially computer systems, can replicate human intelligence and behave intelligently is called artificial intelligence. Artificial intelligence focuses on the theory and practice of self-developing systems in science and engineering that exhibit features associated with human intelligence behaviours, which examines these fields from both a theoretical and practical perspective. In general, artificial intelligence systems take in vast amounts of labelled training data, search through the data for correlations and patterns, and then use these patterns to make predictions about the state of the world in the future. Consequently, artificial intelligence systems are now more adaptable, capable of thought, and generic. A chatbot, for instance, can learn to have lifelike conversations with people by being fed examples of text chats between people, just as an image recognition programme can learn to recognise and characterise objects in photographs by looking at millions of examples of those objects in photographs. In an artificial intelligence system, there is an agent and its environment. The agent might be a person or a robot, but either way, it must be able to monitor its environment and take action based on what it finds. A graphical representation of agents, sensors and effectors is presented in Figure 1. Intelligent agents are required to have the ability to formulate and realise their objectives. In traditional planning problems, the agent can determine the results of its actions by assuming that it is the only system operating in the world at the time. If the agent is not the sole actor, then they need to have the ability to reason when faced with uncertainty. It is necessary to use an agent capable of analysing its environment, making predictions about future events, evaluating those predictions, and modifying itself accordingly. Natural language processing (also known as NLP), speech and face recognition, the prevention of fraud, AI-powered chatbots, applications of A.I. in astronomy, healthcare, and finance, and many more are just some of the enormous applications of artificial intelligence. Consider the following: if someone takes a photograph and wishes to make it look more creative, for example, like a painting done by a cartoonist, how can a computer automatically accomplish this? The image-to-image translation is the name given to this type of research activity, which can be easily accomplished with the help of artificial intelligence[1, 2].

The three capabilities of learning, reasoning, and self-correction, depicted in Figure 5, are the primary focuses of artificial intelligence programming.
In the realm of learning, the primary focus is on amassing information and developing concrete guidelines for processing

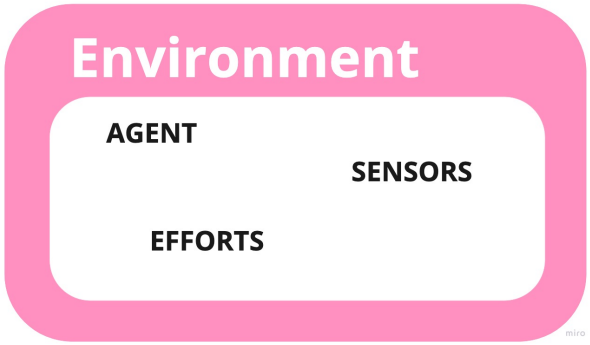

Figure 1: Pictorial Overview Of AI System

it into actionable knowledge. Algorithms are the instructions given to computing equipment to instruct it on how to carry out a specific activity. These instructions are precise.

Reasoning: This area's primary focus is selecting an appropriate algorithm for a given problem.

Self-correction: This area focuses primarily on gaining knowledge from previous errors and determining the most effective solution or available algorithm.

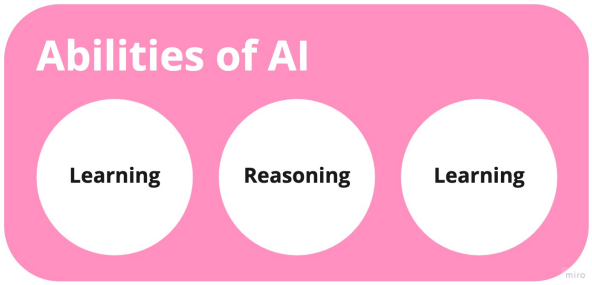

Figure 2: Abilities Of AI

2.1.1 Importance of AI

The rise of artificial intelligence has recently revolutionized the field of computer science and computing. This term's popularity has increased due to recent advancements in artificial intelligence and machine learning. In the field of artificial intelligence, also referred to as machine learning, machines are tasked with performing routine tasks and are conceived to be more intelligent than people. [3] Since the 1990s, quantitative machine learning and network-based machine learning have been two of the most popular types of machine learning. They are designed to acquire new knowledge, adjust to their environment, and carry out tasks more rapidly than humans. Because of advancements in robotics and Internet of Things device integration, robots now have a higher capacity for thought and work than people do. Robots now outperform people in terms of intelligence and cognitive capacity.

1. For a very long time, people have understood the significance of artificial intelligence and its subsequent components, and in order to make the world a better place, they have been employing aspects of artificial intelligence as methods and tools. In addition, using these high-tech devices does not call for any specific training; all we can do is take a brief look around to see that artificial intelligence has probably simplified most of the work.
2. It is essential because it makes day-to-day activities less complicated. These technologies, which primarily benefit people, significantly reduce the effort humans must put in. They are capable of being automated, and manual intervention is the last thing that can be expected or sought during the operation of parts that incorporate this technology.
3. These tools are helpful and efficient because they speed up processes while ensuring accuracy and precision in their results. These technologies and applications not only have a place in our regular and everyday lives,

but they also help make the world more error-free. It influences other fields and is significant in several other fields as well.

It is possible to differentiate between four distinct types of artificial intelligence, beginning with the task-specific intelligent systems in widespread use today and progressing up to the sentient systems that do not yet exist. The following are the various categories:

1. **Reactive machines:** These A.I. systems are tailored to a specific task and do not have a memory. One example is the chess programme Deep Blue from IBM, which beat the human champion Garry Kasparov, in the 1990s. Deep Blue is capable of recognizing the pieces on a chessboard and making predictions, but because it does not have a memory, it is unable to use the experiences it has had in the past to inform the choices it makes in the present and the future.
2. **A limited memory:** These artificial intelligence systems have memories that enable them to draw on the past to guide their actions in the present. It plays a role in the process of decision-making for autonomous vehicles.
3. **Theory of mind:** The field of psychology is responsible for popularising this term. When referring to artificial intelligence (A.I.), this phrase suggests that the technology would be socially intelligent enough to identify feelings. This type of artificial intelligence will be able to predict human behaviour and infer human intentions, a capability that A.I. systems need to possess to become indispensable members of human teams.
4. **Self-awareness:** Artificial intelligence programmes are conscious because they know who they are. There is currently no artificial intelligence that can create machines that are self-aware and aware of their conditions.

2.2 Machine Learning

There is much misunderstanding surrounding the terms "AI" and "ML." the concept of machine learning refers to situations in which a computer can teach itself new information without being given explicit instructions [4]. It is not simple to turn machines into thinking devices, and developing powerful artificial intelligence is only possible by teaching computers to comprehend everything in the same way people do with machine learning. One of the applications of ai is machine learning, which allows computers to teach themselves new skills and improve based on their own experiences without the need for any programming, the same way a human brain learns and comprehends new information.
The primary objective of machine learning is to enhance the learning capabilities of computers and automatically alter their behaviour without the need for human involvement. The first step in machine learning is to look for patterns in the data so that the computer may draw conclusions based on the example given. Similarly, the machine receives information from input, often known as data or training data. A definition that is significantly more succinct [5] is as follows:
(Definition of the [generic] learning problem): "It is said that a computer programme learns from experience E concerning some class of tasks T and performance measure P if its performance on a task in T, as measured by P, increases with experience E." This sentence explains how computer software can learn from its past experiences.

2.2.1 Importance of Machine Learning

The thought of machine learning has been around for a while. Machine learning is a field that studies the research and creation of algorithms that can learn from and predict data. Arthur Samuel, an IBM computer scientist and pioneer in artificial intelligence and computer games, is credited with coining the term "machine learning." Samuel created a checkers-playing computer programme[6]. The more the programme was used, the more it used algorithms to forecast outcomes and learn from experience.
Because it can solve issues at a speed and scale that cannot be matched by the human mind alone, ML has shown to be helpful. Machines can be trained to recognise patterns in and relationships between incoming data by putting large amounts of processing power behind a single activity or several focused tasks. This allows machines to automate repetitive tasks.

1. **Data Is Key:** The success of machine learning depends on its underlying algorithms. ML algorithms create a mathematical model from sample data, also called "training data," to make predictions or choices without being taught. This can highlight patterns in the data organisations can utilise to enhance decision-making, maximise productivity, and collect meaningful data at scale.
   A.I. Is the Goal: A.I. systems that automate workflows and find solutions to data-based business challenges on their own are built on top of machine learning (ML). It enables businesses to supplement or replace

specific human competencies. Chatbots, self-driving cars, and speech recognition are typical machine learning applications you could encounter in daily life.

2. **Data security:** Data security flaws can be found by machine learning models before they result in breaches. Machine learning algorithms can forecast future high-risk activities by considering the past, allowing for proactive risk mitigation. Finance: Banks, trading brokerages, and fintech companies use machine learning algorithms. Companies automate trading and offer investors financial advising services. Bank of America is using a chatbot named Erica to automate customer service.
3. **Healthcare:** Massive healthcare data sets are analysed using machine learning (ML) to enhance patient outcomes, uncover new treatments and cures faster, and automate repetitive tasks to reduce human error. For instance, IBM's Watson employs data mining to give doctors the information they may use to tailor patient care.
4. **Fraud Detection:** The financial and banking industries are using A.I. to automatically evaluate a massive volume of transactions and identify fraudulent behaviour in real-time. According to technology services company Capgemini, fraud detection systems that use machine learning and analytics reduce the time required for fraud investigations by 70% and increase detection accuracy by 90%. Retail: To create A.I. recommendation engines that make appropriate product suggestions based on customers' prior product selections and historical, regional, and demographic data, A.I. researchers and developers are leveraging machine learning (ML) algorithms.

Machine learning has definite advantages for A.I. technology. There are three main ML training methods shown in Figure 3.

1. **Supervised Learning:** Supervised machine learning algorithms use labelled examples of applying what they have learned in the past to new data. The learning method creates an inferred function to forecast output values by examining a known training dataset. After adequate training, the system may provide targets for any new input. To discover problems and correct them, model, as necessary, it can also compare its output with that which is proper and intended.
2. **Unsupervised Learning:** These machine learning techniques are utilised when training data is neither categorised nor labelled. Unsupervised learning is the learning of how computers can derive a function using unlabelled information to describe a hidden pattern. The program never has a particular notion of the correct output; instead, it infers datasets' results. Unsupervised machine learning techniques are utilised when training data does not contain classification or labelling.
3. **Reinforcement Learning:** Reinforcement learning algorithms interact with their surroundings by taking actions and identifying successes or failures. Trial-and-error learning and delayed rewards are two of reinforcement learning's most essential features. With the help of this technique, machines and software agents may automatically select the best course of action in a given situation to enhance performance. The reinforcement signal, or direct incentive feedback, is required for the agent to decide which behaviour is preferable.

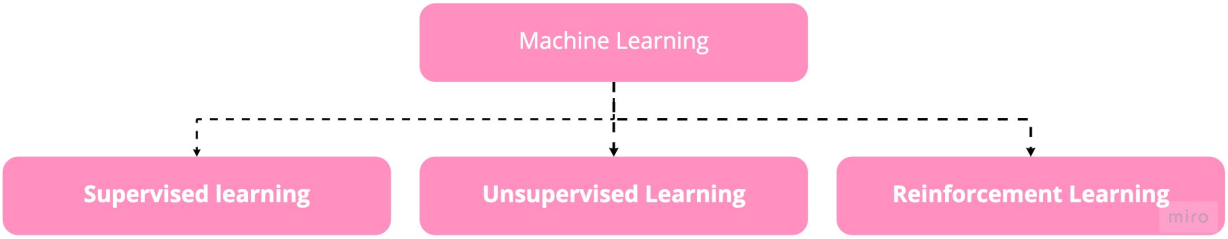

Figure 3: Types Of Machine Learning Methods

Despite all of its drawbacks, machine learning is still essential to the success of A.I.. However, a different approach to A.I. that addresses its flaws, such as the "black box" problem that arises when robots learn unsupervised, will be necessary for this achievement. That strategy is symbolic A.I., sometimes known as a rule-based approach to data processing[7]. A knowledge graph, an open box used in a symbolic method, is used to define concepts and semantic relationships. Hybrid A.I., which combines ML and symbolic A.I., enables A.I. to comprehend language in addition to data.
As the task is to handle the same type of data, there are three most common tasks in machine learning:

1. **Classification:** Based on training data, the Classification algorithm is a Supervised Learning technique used to categorise new observations. The Classification algorithm uses labelled input data because it is a supervised learning technique and comprises input and output information. In classification, a programme uses the dataset or observations provided to learn how to categorise new observations into various classes or groups. For instance, cat or dog, yes or no, 0 or 1, spam or not spam, and many more. Targets, labels, or categories can all be used to describe classes. In contrast to regression, classification's output variable is a category rather than a value, such as "Green or Blue," "fruit or animal," and many more examples.
2. **Regression:** It is a supervised machine learning task that uses a group of connected features to predict the label's value. The label is not limited to a small selection of values like in classification jobs and can be of any real value. Regression methods predict how the label will change when the values of the linked features vary by modelling the dependency of the label on those features. Regression algorithms take a series of instances with labels for known values as their input. A function that you may use to forecast the label value for any fresh collection of input information is the result of a regression process—for example, the prediction of house prices with the help of different attributes like bedrooms, kitchens and many more.
3. **Clustering:** One of the unsupervised machine learning tasks that groups data instances into clusters with related properties. Additionally, clustering can be utilised to find connections in a collection that browsing or straightforward observation might not logically reveal. A clustering algorithm may have different inputs and outputs depending on the methodology selected. You can choose a method based on distribution, centroid, connectedness, or density, for example, identifying categories of hotel guests based on their habits and preferences.

The statistic used to assess the capabilities of machine learning algorithms is called P (Performance). A model's performance is always evaluated in terms of accuracy for classification and clustering tasks, and it is precisely the percentage of accurate samples to the entire sample. Calculating MSE (Mean- Square Error) of the fit line and actual data points for regression is a typical approach to evaluating the model's quality. The MSE formula is:

$$MSE = 1/n \sum_{n} (yi - f(xi))^2 \tag{1}$$

A portion of the dataset is used as a test set to complete the performance evaluation, and a test set is used to evaluate a model's performance after training.
Aside from the test set, another portion of the complete data set is frequently retrieved as a validation set in current ML projects. Cross-validation is commonly implemented using the validation set[8]. Cross-validation is used to avoid over-fitting the model by including non-training data throughout the training process.
There are two main challenges in ML: underfitting and overfitting[9]. These two ideas relate specifically to the model's capacity for generalisation. A model is said to be generalised if it can still produce good results with unknown input. The aim of ML is also to achieve good generalizability. A model looks underfitting if it does not exhibit good generalisation in the training set. Overfitting is evident when a model performs well in generalisation on the training set but poorly on the test set. Figure 4 demonstrate the perfect example of underfitting, overfitting and most suitable fitting.

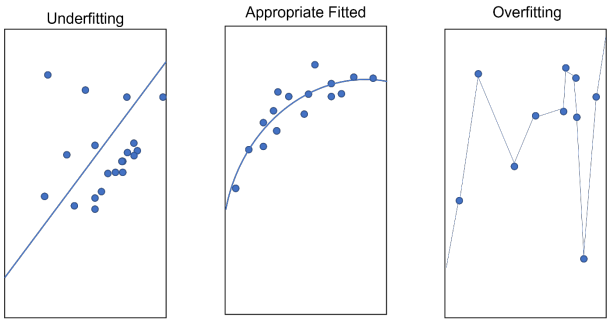

Figure 4: Examples of under-fitting, appropriate and over-fitting

2.3 Deep Learning

Deep Learning teaches computers to learn by mimicking how humans acquire knowledge in their everyday lives. Deep learning is an essential technology component that enables driverless automobiles to recognize stop signs and

differentiate between pedestrians and lampposts [10]. It is necessary to have voice control capabilities built into consumer gadgets, such as hands-free speakers, tablets, televisions, and smartphones. For a while, there has been a great deal of interest in deep learning, and there is a strong reason for this. It is producing results that were previously impossible to achieve. Using a " deep learning technique," a computer model can be trained to carry out categorization tasks directly from images, text, or voice. Models trained using deep learning can achieve levels of precision comparable to or even superior to humans. When training models, a sizeable amount of data, either labelled or unlabeled, as well as multi-layered neural network designs, are utilized. The level of recognition accuracy that can be achieved with deep learning is higher than ever. This is vital for ensuring that consumer devices live up to customers' expectations for safety-sensitive applications like driverless cars. As a result of recent advancements, deep learning is now capable of doing better than humans in specific tasks, including classifying objects in photographs, processing natural languages, medical diagnosis, and prediction [11].
Even though the concept of deep learning was first proposed in the 1980s, its usefulness has just lately begun to emerge for two primary reasons:

1. In order to perform deep learning, you will need a large number of labelled data sets. For example, the development of a self-driving automobile requires taking millions of photographs and countless hours of video.

2. Because deep learning requires a significant amount of processing power, it is effectively supported by the parallel design of high-performance GPUs. Combined with cluster or cloud computing, it is possible for development teams to reduce the time required for training a deep learning network from weeks to hours or even fewer.

Machine learning algorithms can generate predictions with the help of structured data that has been labelled. The model's input data are utilized to determine particular characteristics, which are subsequently catalogued using tables. This does not necessarily suggest that it does not use unstructured data; instead, it shows that if it does, the data will generally undergo some form of pre-processing to be organized in a structured fashion. Deep learning eliminates the need for some of the typical data pre-processing steps involved in machine learning. As a result of these algorithms' ability to process unstructured text and visual data and automate feature extraction, the demand for human specialists is significantly reduced. Refer to Figure 5 for a better understanding. Imagine for a moment that we had a collection of pictures of different animals kept as pets and that we wanted to sort them into categories such as "cat," "dog," "hamster," etc. The use of deep learning allows for the development of algorithms that can determine which traits, such as ears, are essential in distinguishing one species from another. This hierarchy of features was painstakingly crafted by a human expert in the field of machine learning. Machine learning and deep learning models can learn in a variety of different ways, in addition to being capable of supervised learning, unsupervised learning, and reinforcement learning. To categorize or make predictions, supervised learning uses labelled datasets; this requires human intervention to label input data appropriately. On the contrary, unsupervised does not require any labelled datasets; instead, it examines the data in search of patterns and organizes the data according to any distinguishing characteristics. A model can learn, through the process of reinforcement learning, to carry out a task in an environment more accurately in order to increase the amount of reward it receives. The deep learning algorithm will then fine-tune and adjust itself for accuracy through gradient descent and backpropagation, enabling it to generate more accurate predictions for a fresh animal shot.
Since neural network topologies are used in most deep learning techniques, deep learning models are also referred to as deep neural networks. The term "deep" refers to the numerous hidden layers and nodes inside the neural network. Deep networks can contain an arbitrary number of hidden layers, but standard neural networks typically have between two and three hidden levels. For training deep learning models, extensive data collections with labels and neural network topologies can automatically learn properties from the data. These also make an effort to mimic the human brain by employing data inputs, weights, and biases in their decision-making processes. Combining these aspects ensures that the elements in the data are correctly identified, categorized, and characterized.
A neural network comprises multiple layers of interconnected nodes, each of which tries to improve the classification or prediction given by the layer behind it. The only layers that are visible in a deep neural network are the input and output layers. After the deep learning model has processed the data in the input layer, it is the output layer responsible for making the ultimate prediction or classification. Calculations are moved from one location to another inside a network called "forwarding propagation."
The term "backpropagation" refers to a method that makes use of techniques such as "gradient descent"[12]. Backpropagation is now the most prevalent approach for training a neural network[13], and it is used to quantify prediction errors. It also modifies the function's weights and biases by travelling back through the layers to train the model. Backpropagation is also used to train the model. Because of forwarding propagation and backpropagation, a neural network can produce predictions and repairs for any errors. The precision of the algorithm consistently increases with time. Deep learning techniques are notoriously tricky to master, and numerous neural networks are available to address a wide range of problems and datasets. Take, for instance:

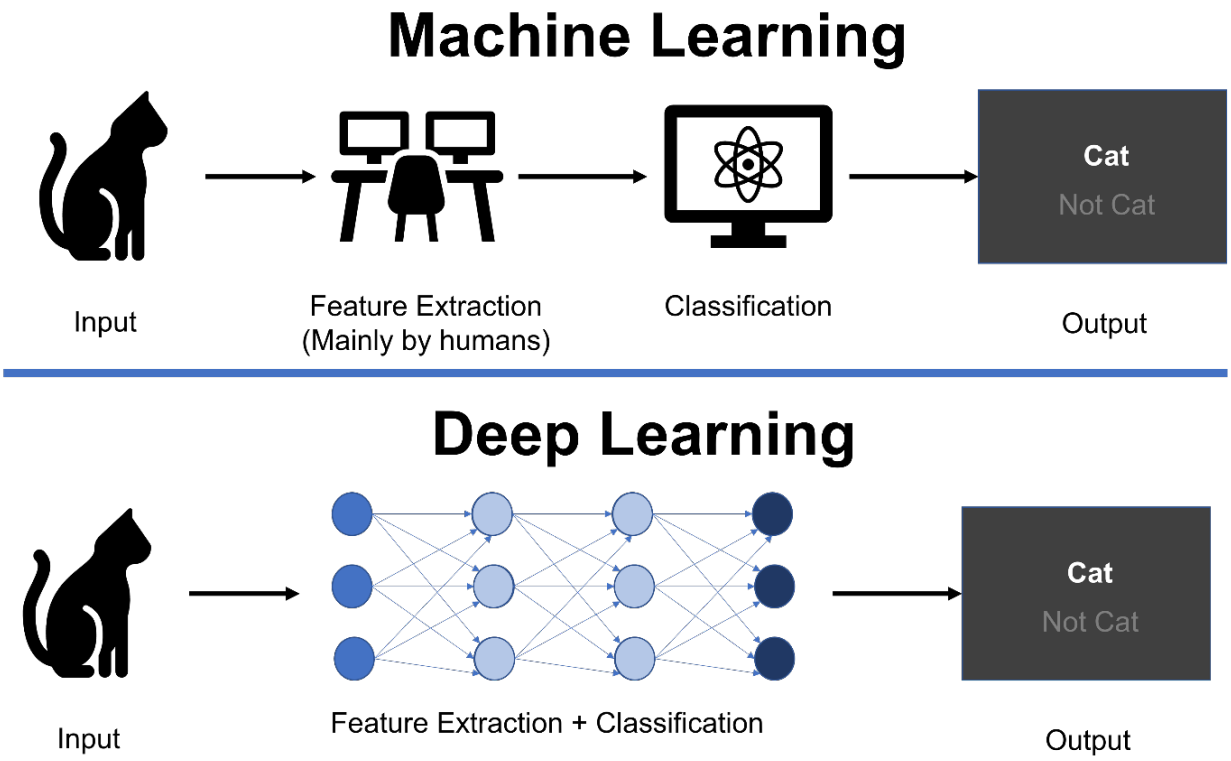

Figure 5: Difference between deep learning and machine learning

1. Convolutional neural networks, also known as CNNs, are a type of artificial neural network primarily used in computer vision and image classification applications. CNNs can recognize patterns and features within an image, enabling them to perform object recognition or detection tasks. One of the essential CNN models for object detection, R-CNN, was published in [14].
2. Recurrent neural networks (RNNs): RNNs are commonly employed in applications involving speech and natural language recognition because they utilize sequentially or time series data.

2.3.1 Deep Learning Methods

There are several different approaches that can be used to construct robust deep learning models. Dropout, learning rate decay, transfer learning, and starting from scratch are some of the approaches included in this category.

1. **Learning rate decay:** It is one of the hyperparameters that determines how big of a change the model goes through in response to the projected error every time the model weights are altered. A hyperparameter is a component that, prior to the learning process, either defines the system or creates the circumstances necessary for it to function properly. It's possible that an inferior collection of weights may be taught, or that excessive learning rates could lead to unstable training processes. Learning rates that are too slow can result in a drawn-out training procedure, with the danger of getting stuck along the way.
   The approach of modifying the learning rate in order to improve performance and shorten the amount of time spent training is referred to as the learning rate decay method. This method is also referred to as learning rate annealing or adaptable learning rates. Among the most easy and well-known modifications of the learning rate that may be made during training are strategies that, over time, cut down the rate at which new information is acquired.
2. **Transfer learning:** Is a method that requires access to the inner workings of a network in order to perfect a model that has already been trained using the method. This method involves refining a model that has been trained. Users begin by contributing fresh data to an already-established network, which may include classifications that were not previously known. When the network has been updated, it will be possible to do new jobs that require more precise categorization abilities. The computation time can be lowered down to

minutes or hours with this method because it requires only a small fraction of the data that is required by other methods.

3. **Training from scratch:** Requires the compilation of a large data set that has been labelled, as well as the construction of a network architecture that is capable of training the model and its features. This tactic may be of substantial use to applications that provide many different output categories as well as to new applications. However, due to the fact that it requires a large amount of data and takes several days or weeks to train, it is typically one of the less popular strategies.
4. **Dropout:** By arbitrarily eliminating units and their connections while the network is being trained, the dropout method is an attempt to solve the problem of overfitting in neural networks that contain a large number of parameters. It has been established that the dropout technique can improve the performance of neural networks when it comes to supervised learning tasks. These tasks include document categorization, speech recognition, and computational biology.

2.4 Convolutional Neural Network

The capacity of artificial intelligence to bridge the gap in competence between humans and machines has seen significant progress in recent years. In order to attain exceptional achievements, experts and hobbyists must concentrate on many aspects of the area. One of these fields is computer vision, but there are many others. Research in this area aims to endow computers with the capacity to perceive and comprehend the world in the same way humans do. After achieving this level of comprehension, they can apply it to a wide range of activities, including image and video recognition, image analysis and classification, media recreation, recommendation systems, natural language processing, and many more. In particular, over time, a Convolutional Neural Network approach has been developed and optimised, which has led to significant advances in the field of computer vision.
The aim of a deep learning neural network known as a convolutional neural network, or CNN for short [15], is to process organised arrays of input such as images. Computer vision extensively uses the most advanced techniques available for various visual applications. Some examples of these techniques are picture categorisation and convolutional neural networks. Additionally, they have seen success in natural language processing for text classification.
Convolutional neural networks are highly good at recognising the patterns in the input image. These patterns include eyes, faces, lines, gradients, and circles. Because of this property, convolutional neural networks are instrumental in computer vision. In contrast to more traditional approaches to computer vision, Convolutional neural networks do not require any preparation and can begin processing a raw image immediately. In order to process input images and recognise more complex components than ever, a convolutional neural network uses convolutional layers. This network organisation is meant to resemble the structure of the human visual brain.
Convolutional Neural Network Design
Due to the sequential nature of their formation, convolutional neural networks have the ability to learn hierarchical features. CNN uses a number of different convolutional layer groups, which are typically followed by activation layer groups, and some of which are subsequently followed by pooling layer groups as the hidden layers. The appearance of each of the layers is depicted in Figure 6. LeNet-5, an early convolutional neural network, is an uncomplicated convolutional neural network that enhances the understanding of core design ideas [16]. The handwritten characters can be read by LeNet.

2.4.1 Convolutional Layer

One of the essential parts of a convolutional neural network is the convolutional layer. A convolutional layer can be considered a collection of convolutional kernels or minimal square templates with a search function that moves through an image looking for patterns. If the specified region of the image follows the pattern of the kernel, the kernel will supply a sizeable positive number. The kernel will return either zero or a lower value if it does not.
Activation functions in CNN: After an image has been passed through a convolutional layer, applying an activation function to the result of that layer's processing is common practice. The nonlinear change we apply to the input signal is referred to as the activation function. This altered output is then passed as input to the layer of neurons or convolution that comes after it. The activation functions most frequently utilised in neural networks are outlined in Table 1.
The ReLU activation function is the one that is utilised in neural networks more commonly than any other function, including all the other functions. ReLU has a significant competitive advantage over other activation functions because it does not stimulate all of the neurons at the same time. This is a substantial advance. The formula for the ReLU function, which can be found up top, illustrates that it brings every negative input down to zero and stops the neuron from being triggered. This can be found in the previous sentence. Since only a few neurons at a time are stimulated, it is highly efficient from a computational point of view. When it is in the positive area, it never reaches the point of saturation. When put into reality, the ReLU activation function is six times faster than the Tanh and Sigmoid activation functions.

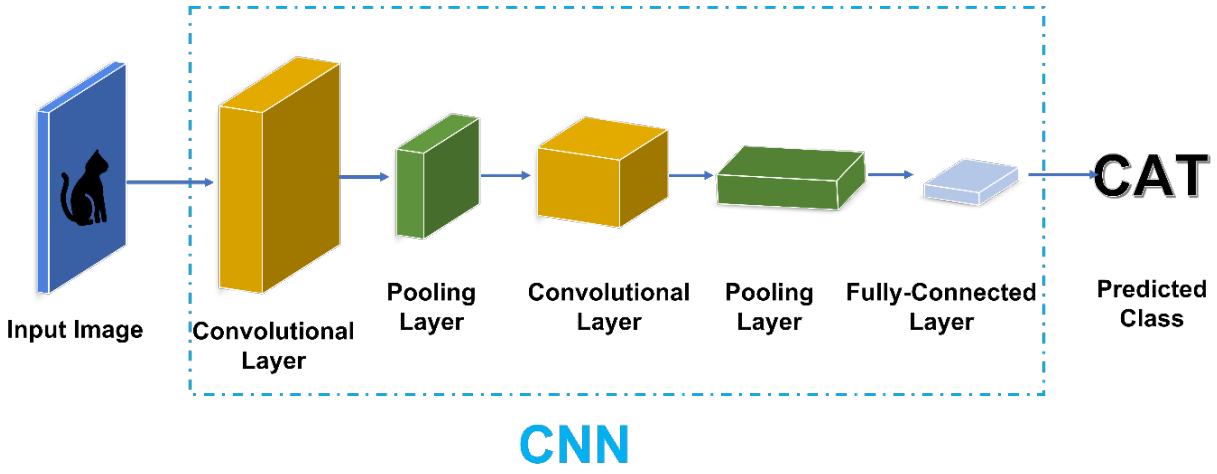

Figure 6: Basic CNN Architecture for classification

Table 1: Summary of Gaze datasets allowing continuous head poses and gaze directions.

SNo	Activation Function	Formula
1	Sigmoid	$\sigma(x) = 1/(1+e)^{-}x$
2	tanh	$tanh(x)$
3	ReLu	$max(0, x)$
4	Leaky ReLu	$max(0.1x, x)$
5	Maxout	$max(w_1^T x + b_1, w_2^T x + b_2)$
6	ELU	$\begin{cases} x & x >= 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

2.4.2 Pooling Layer

In a CNN design, the Pooling layer may be seen between the Convolution layers. This layer minimises the amount of parameters and computations in the network, preventing overfitting by gradually shrinking the network's spatial size. Some of the pooling techniques are described below:

1. **Average Pooling:** Average Pooling is a pooling technique that computes the average value of a feature map's patches and utilises it to build a reduced (pooled) feature map. It is typically employed following convolution layers. It adds a bit invariance, which means that translating the picture by a small amount has little effect on the results of most pooled outcomes. An example of the average pool is shown in figure 7.

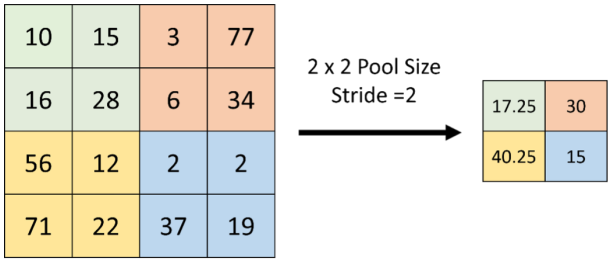

Figure 7: Example of Average Pooling

2. **Maximum Pooling:** As the name implies, Max-pooling will extract only the greatest from a pool. This is accomplished by sliding filters through the input; the highest parameter is taken at each step, and the remainder is dropped. The network is compressed as a result of this. An example of max-pooling is shown in figure 8.

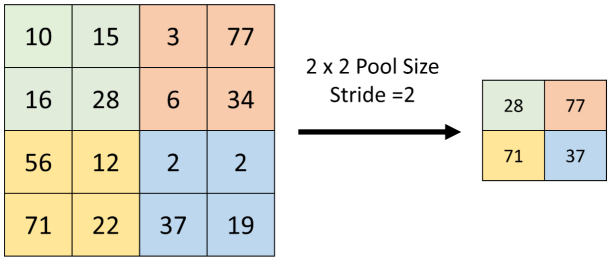

Figure 8: Example of Max Pooling

However, unlike the convolution layer, the pooling layer does not modify the depth of the network; the depth dimension stays unchanged.

2.4.3 Fully Connected Layer

The neurons in this layer are entirely connected to the previous levels' activations. Their activations can thus be determined using matrix multiplication and a bias offset. This is the final stage of a CNN network. The CNN is composed of both hidden and fully-connected layers (s). Many advanced neural network models are built on CNN's convolution structure. A few well networks are as follows:
VGG Net[17], AlexNet[18], ResNet, and U-Net[19].

2.5 Segmentation

Most of us with a reasonable level of experience with computer vision problems will be able to recognise two basic types of problems. Image Categorisation and Image Detection on the peripheral of these topics - Image classification is a type of problem in which we are concerned with the existence of a picture in a scene, followed by image detection and localisation, which defines the region where a given object is located and draws a boundary box/ellipse around it. However, there is a big brother to them which is image recognition.
Image Segmentation is the most challenging and perhaps most helpful class of issue among the three. So, in straightforward terms, image segmentation is a problem in which each pixel is classified into one of the classes of entities in a given scene.
Image segmentation is classified into two types:

1. Semantic segmentation
2. Simultaneous Detection, also known as instance aware segmentation.

As you might have guessed from their names, both types are the same, except that Semantic segmentation is only concerned with categorising each pixel. In contrast, Instance Aware segmentation is concerned with identifying the individual instances of each object. For example, if there are three cats in a picture, Semantic segmentation is involved with classifying all three cats as one instance. In contrast, Instance segmentation might identify each of them individually.
Figure 10 shows the illustration of the difference between semantic and instance segmentation.
Some of the Deep Learning based image segmentation models are shown in table 2. Our main focus is on Encoder-Decoder-based models, as we will be using this technique for our project

2.5.1 Encode-Decider Models

It is made up of two parts: an encoder and a decoder. A convolutional layer is used in an encoder, whereas a deconvolutional network is used in a decoder to build a map of pixel-wise probabilities using the input feature vector. The two most common architectures used in healthcare picture segmentation and other types of segmentation challenges are U-Net and V-Net[19, 20]. U-Net is mainly used for image segmentation in biological microscopy. However, UNet-based models are increasingly employed for segmentation in a variety of areas such as human parsing, remote

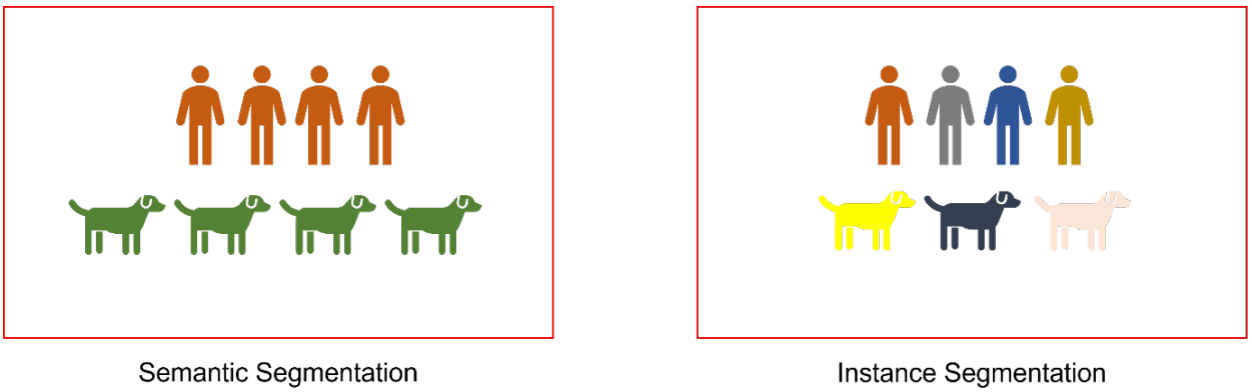

Figure 9: Semantic Segmentation Vs Instance Segmentation

Table 2: List of image segmentation models

SNo	Image Segmentation Models
1	Fully Convolutional Networks
2	Encoder-Decoder Based Models
3	Convolutional Models with Graphical Models
4	Multi-Scale and Pyramid Network Based Models
5	R-CNN Based Models (for Instance Segmentation)
6	Dilated Convolutional Models and DeepLab Family
7	Generative Models and Adversarial Training
8	CNN Models with Active Contour Modelsy
9	Recurrent Neural Network Based Models
10	Attention-Based Models

sensing, automotive classification, and many more[21, 22, 23]. It learns from annotated photos using data augmentation techniques. The U-Net architecture is divided into a shrinking path and a symmetric extending path for collecting context and enabling exact localisation.

From Figure 10 we can see that UNet architecture[19] is divided into two parts: i) Encode and ii) Decoder. Encoder(Left Side): It comprises two 3x3 convolutions applied repeatedly. Following each conv is a ReLU and batch normalisation. The spatial dimensions are then reduced using a 2x2 max pooling operation. At each down sampling stage, we double the number of feature channels while cutting the spatial dimensions in half. Decoder(Right Side): Each step in the long path comprises an up-sampling of the feature space accompanied by a 2x2 transpose convolution, which reduces the number of feature channels by half. In addition, we have a concatenation with the equivalent feature space from the contracting path and a 3x3 convolutional filter (each followed by a ReLU). A 1x1 convolution is employed in the final layer for mapping the channels to the appropriate number of classes.

2.6 Residual Networks-ResNet

Deep Residual Network was undoubtedly the most groundbreaking achievement in the computer vision/deep learning area, in the previous several years, following the renowned success of AlexNet [18] at the ImageNet Large Scale Visual Recognition Challenge 2012 classification challenge. ResNet allows you to train many layers while still achieving impressive results. Using its excellent representational ability, it has improved the performance of numerous applications for computer vision other than image classification, including object identification and face recognition. Given enough capacity, the universal approximation theory stated that a feed-forward neural network with a single layer might represent any function. However, the layers may be huge, and the network could overfit the data. As a result, there is a widespread consensus in the academic world believe our network architecture requires more depth.
Since AlexNet, cutting-edge CNN architecture has gotten deeper and deeper. AlexNet only had five convolutional layers, whereas the VGG net[17] and GoogleNet (also known as Inception v1) [24] featured 19 and 22 layers, respectively.
However, just stacking layers together does not work to increase network depth. Deep networks are challenging to

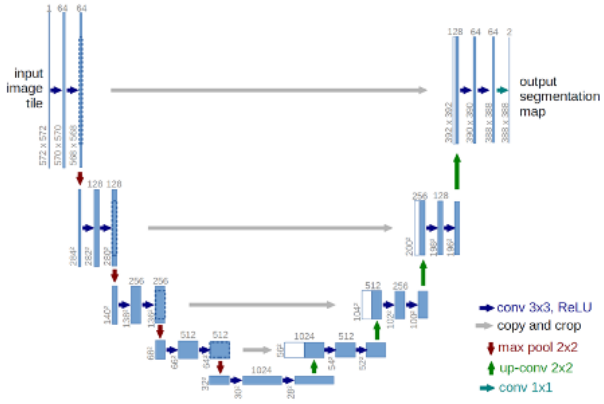

Figure 10: UNet Architecture

train due to the well-known vanishing gradient problem. As the gradient is propagated back to older layers, repeated multiplication can lead the gradient to become infinitely small. As a result, as the network penetrates deeper, its performance becomes saturated or even degrades rapidly.
Before ResNet, there were numerous approaches to dealing with the vanishing gradient problem; for example, [24] inserts an extra loss in a central portion as additional supervision, and none seemed to solve the issue. The main concept of ResNet is to introduce a so-called "identity shortcut link" that bypasses one or even more layers.
ResNets can vary in size based on how big each layer of the model is and the number of layers it does have.

3 Eye Gaze Tracking: Review

3.1 Datasets

Dataset plays a very crucial role in estimating gaze with good accuracy. Today, a number of datasets are available publicly, which contains a wide range of images under different environments and constraints. Some of the widely used dataset are: MPIIGaze[25], eyediap[26], Columbia[27], GazeCapture[28], TabletGaze[29] etc. Some of the datasets allows continuous head poses and gaze directions. Gaze360[30] datasets contain over 172,000 images from 238 subjects looking at different gaze directions. Data was collected in 5 indoor (53 subjects) and 2 outdoors (185 subjects) locations over 9 recordings sessions with labelled 3D gaze across a wide range of head poses and distances. MPIIGaze[25] dataset contains 213000 images from 15 subjects collected via laptop under different illumination conditions. This dataset is applicable for both 2D and 3D gaze estimation. Eyediap[26] dataset contains 94 videos from 16 subjects. It contains 3 minutes video sequences of subjects looking at fixed and floating targets taken from RGB-D (standard vision and depth) cameras. This dataset is designed in such a way that it is least affected by head pose variations, changes in ambient and sensing conditions, fixed and floating targets, person specific variations. RT-Gene[31] contains 123,000 images from 15 subjects. It allows automatic annotation of ground truth gaze and head poses labels of subjects under free-viewing conditions and large camera to subject distances. However, it is only applicable for 3D gaze estimation. This dataset requires separate eyetracking glasses along with RGB-D cameras and motion capture cameras to collect data. It also makes use of GAN to remove eye tracking glasses after collecting ground truth. The dataset contains RGB images at 1920x1080 resolution and depth images at 512x424 resolution. Deng and Zhu [32] introduced a novel dataset that contains 240,000 images from 200 subjects. This dataset is collected in such a way that it can be used for both 3D and 2D gaze estimation, should be device-independent, and contains full coverage of head poses and gaze estimation under varying illumination conditions. In the data collection setup, two types of targets (Head pose targets and eyeball targets) are displayed to participants in order to guide their head pose and eyeball movements respectively. 12 cameras are used to capture a wide range of head poses and to provide 3D gaze annotations. GazeCapture[28] dataset contains 2.4 million images from 1474 participants. This dataset is collected using crowdsourcing by using mobile phones and tablets in different orientations. This dataset provides gaze direction as a pixel locations on the screen and distance from the camera. Dubey et al. [33] introduced a novel dataset that contains 1,54,251 images of 100 different subjects from YouTube videos. Different types of videos are downloaded from a creative common section of YouTube. After that every third frame is considered for dataset creation. Zhang et al. [34] introduced a novel dataset with an objective that it can be used in multiple devices(mobile phone, tablet, laptop, desktop, smart TV). A total of 11080 samples are collected from 22 participants. The camera resolution used for each device were: 1440 x 2560 in case of mobile phones, 2560 x 1600 pixels for tablets, 1920 x 1080 pixels for laptop and smart TV and 1920 x 1200 pixels for desktop

Table 3: Summary of Gaze datasets allowing continuous head poses and gaze directions.

Referances	Dataset Name	Size	Head Poses	Gaze Direction	Subjects	Description
Kellnhofer et al. [30]	Gaze 360	172k images	Continuous	Continuous	238	Data was collected in 5 indoor (53 subjects) and 2 outdoor (185 subjects) locations over 9 recording sessions with labelled 3D gaze across a wide range of head poses and distances
Zhang, Sugano, Fritz, et al.[25]	MPIIGaze	213k images	Continuous	Continuous	15	Collected by laptops for daily life illumination conditions. Applicable for 2D and 3d gaze estimation.
Funes Mora et al.[26]	Eyediap	94 videos	Continuous	Continuous	16	Contains 3 minutes video sequences of subjects looking at fixed and floating targets
Krafka et al.[28]	GazeCapture	2.4M	Continuous	Continuous	1474	Collected by mobile devices via crowdsourcing. Only for 2D gaze.
Fischer et al[31]	RT-GENE	123K	Continuous	Continuous	15	For 3D gaze estimation only. Requires separate eyetracking glasses along with RGB-D camera and motion capture camera
Park, Aksan, et al.[35]	EVE	4.2K videos	Continuous	Continuous	54	For both 2D and 3d gaze estimation. Video frames are captured in 1920x1080 pixels.
Deng and Zhu[32]	-	240K images	Continuous	Continuous	200	Dataset is device independent. Can be used for both 2D and 3D gaze estimation.
Dubey et al.[33]	-	154K images	Continuous	Continuous	100	Videos were downloaded from YouTube creative section. Then third frame is considered for data creation
Lian et al.[36]	-	165K	Continuous	Continuous	218	Largest RGBD gaze dataset in terms of participants
Zhang, Huang, Sugano, et al.[34]	-	11080 samples	Continuous	Continuous	22	Can be used in Mobile/ Laptop/ Desktop/ Tablet devices.

computer. Table 3 summarizes gaze datasets that allows continuous head poses and gaze directions.
Some of the datasets are still limited in gaze direction, head poses and illumination conditions. Columbia [27] dataset contains 6000 images from 58 participants. It consists of only 5 head poses and 21 gaze directions per head poses. Subjects were asked to fix their head on a chin rest while collecting data. The data from each subjects consists of 5 different head poses, 3 eye vertical movements and 7 eye horizontal movements contributing a total of 105 images per subject. The dataset is very diverse in nature as it contains 24 females and 32 males of different age groups. Rice TabletGaze[29] dataset contains video recordings of 51 subjects consisting of 39 males and 12 females of different age groups. The subjects were looking at 35 points distributed among 5 rows and 7 columns. The device used for data collection is Samsung Tab S 10.5. Four recordings of four postures (Sitting, Slouching, Standing, and Lying) were collected from each participants resulting in a dataset consisting of 16 videos per subject. Unlike Columbia dataset here participants were not asked to remain in a fixed head pose. However, gaze directions were limited to 35 directions only. Wood and Bulling [37] presented a novel dataset consisting of 8 participants aged between 20 to 27. Dataset was collected on a 11-inch tablet with a quad-core 2 GHz processor running on windows 8. Each participant was asked to look at 9 pre-defined locations distributed on a 3 x 3 grid pattern. Although participants were allowed free head movement, distance between the eyes of the participants and device was fixed to 20 cm. Tablet was held in reverse orientation with camera at the bottom. Xia et al. [38] presented a dataset that contains 200 frames from 550 participants resulting in a total of approximately 110,000 images. The participants were in the range of 20-35 years of age. The data was collected using a Samsung phone with a screen resolution of 2220 x 1080 pixels. A total of 25 fixed gaze points were presented in screen and participants were asked to look at these points. Participants were allowed free head movements and distance between participants and screen was not fixed and varies from 25 to 60 cm.

3.2 Results and Discussions

In this section we have discussed about the results obtained in various works. gaze tracking accuracy measures are reported in different ways for e.g. angular accuracy in degrees [39, 40, 41, 42], distance accuracy in cm/mm[43, 44, 45, 46, 47], distance in pixels and gaze estimation accuracy in percentages[48, 30, 33].
George 2016 [48] Proposed a real time classification framework for eye gaze direction for predicting 7 eyes accessing cues (EAC) classes. He proposed two different methods, first one where landmark detection is carried out using geometrical relations and the second method where ensemble of randomized tree approach (ERT) is used for landmark detection. The first method gave an accuracy of 81.37 percent and the second method accuracy was 86.81 percent. The best result was obtained with the second method when combined with the CNN architecture discussed previously in this work.
Dubey 2019 [33]proposed a method which estimates eye gaze mapping using unsupervised learning methodologies.

For training the model, he initialized weights using ‘glorot normal ‘distribution and then trained it using stochastic gradient descent optimizer with a learning rate of 0.001. Categorical cross entropy is used as a loss function to train the network. He achieved an accuracy of 91.5 percent on the proposed dataset.
Some works estimated accuracy in terms of degrees. Park 2018 [39] introduced a novel deep neural network architectures to estimate gaze using single eye input. He trained network using ADAM optimizer with a batch size of 32 and a learning rate of 0.0002. He performed the evaluation on three datasets and achieved an accuracy of 4.5 degrees in MPIIGaze[25] dataset, 10.3 degrees on eyediap[20] dataset and 3.8 degrees in Columbia dataset. Park 2019[41] presented a deep learning gaze estimation methods which can achieve high accuracy requiring very few calibration samples. The entire network was trained using stochastic gradient descent optimization method and it achieved an accuracy of 3.18 degrees in Gazecapture[krafka2016eye] dataset and 3.42 degrees in MPIIGaze[25] dataset. Zhang 2017 [25]proposed a novel method based on multimodal convolutional neural network. From the input image obtained from the RGB camera, facial landmarks and face image are detected. A convolutional model is then used to learn a mapping from features obtained to 3D gaze directions. The proposed method obtained an accuracy of 10.8 degrees improving the state of art by 22 percent from mean error of 13.9 degrees to 10.8 degrees. Zhang 2016 [42] proposed a method that only takes full face image as input. It makes use of spatial weights to suppress and enhance performance in different facial regions. The proposed method achieved an accuracy of 4.8 degrees in MPIIGaze dataset and 6 degrees in eyediap dataset resulting in improvement of accuracy of up to14.3 percent on MPIIGaze and 27.7 percent on eyediap dataset for 3D gaze estimation. Cheng 2020[49] proposed a coarse to fine strategy to estimate gaze. Two convolutional neural networks are designed to estimate gaze, one to extract coarse features from eye images and predict basic gaze direction. Another convolutional neural network is to extract fine features. Then results from both networks are combined to estimate gaze. The results were compared with iTracker[28], RT-Gene[31]. The prosed method achieved an accuracy of 4.1 degrees in MPIIGaze dataset and 5.3 degrees in eyediap dataset and outperformed other appearance based methods. Palmero 2018 [50] proposed a method to estimate gaze using a multimodal recurrent convolutional neural network. He achieved an accuracy of 5.1 degrees in static head pose conditions and 6.2 degrees in moving head pose conditions achieving an improvement of 14.6 percent over the state of art methodologies like MPIIGaze method[25] on eyediap dataset. Park 2020[35] proposed a architecture for end to end video based eye tracking. The proposed method achieved a 28 percent improvement in point of gaze estimation resulting in 2.49 degree error. Li 2018[51] proposed a combined gaze tracking algorithm where a convolutional neural network is utilized to remove blinking images and predict a coarse gaze direction. Next, a geometric model is used for accurate gaze tracking. The proposed method achieved a gaze accuracy of 0.53 degrees. Wood 2014 [37] proposed a model based approach for gaze estimation that runs on unmodified tablets. First, eye region of interest and elliptical outline is obtained using robust model-fitting method. The 2D ellipses are then back projected to 3D to find optical axes. Point of gaze is then estimated using the intersections between optical axes and screen. The proposed method achieved an accuracy of 6.88 degrees.
Some works estimated accuracy in terms of distance in cm/mm between the actual and predicted gaze direction. Chang 2019 [44] proposed a method which utilizes saliency information to estimate gaze direction without explicit user calibration. The proposed method achieved an error of 3.3 cm resulting an accuracy improvement of about 24 percent over existing methods like iTracker[28] by 24 percent. He 2019 [45] proposed a on device few shot personalization method for 2D gaze estimation. The proposed method can achieve better accuracy using very few calibration points and achieved 24.26 percent better accuracy compared to other existing methods. The method achieved an accuracy (measured by mean error) of 1.37 cm on mobile devices and 2.1 cm on tablets. Krafka 2016 [28] proposed a deep convolutional neural network for estimating gaze, achieving an error of 1.04 cm on mobile devices and 1.69 cm on tablets respectively. The proposed method achieved a significant reduction in error as compared to other approaches like MPIIGaze[25]-3.63 cm, Tabletgaze[29]-3.17 cm and Alexnet[42]-3.09 cm respectively.
Lack of homogeneity can be observed in performance evaluation among several approaches. While some works estimated accuracy in percentage, others have measured accuracy in terms of distance or degrees. This variation makes inter-comparisons between different works improbable. There is need of development of standard methodologies for evaluating performances of different methods.

3.3 Device and Platforms

In this section, we have discussed about the devices which are used to capture data. Also user platforms where eye gaze tracking is incorporated are discussed. Mainly three platforms are used: computers, handheld devices, and head-mounted devices.
The majority of gaze estimation systems makes use of a single RGB camera to gather data, while some systems makes use of different camera settings, e.g., using multiple cameras to gather multi-view images[30], using infrared (IR) cameras to tackle low illumination condition[51] and using RGBD cameras to collect the depth information[31, 36].
Kellnhofer et al. collected data using setup built on a Ladybug5 360° panoramic camera placed on tripod [30]. To build the dataset, they have used AlphaPose[52]to detect head key point positons and participant’s feet from each camera unit independently. Wang et al. collected data through 4 different cameras in different perspective to have dataset more

Table 4: Summary of Gaze Estimation Methods applicable to Desktops

Referances	Accuracy	Architecture	Dataset	Image Resolution	Input
George and Routray [48]	89.81%	Own CNN architecture	Eye Chimera	42x50	Two eyes
Kellnhofer et al. [30]	51%	Bidirectional LSTM	Gaze360,Own Dataset	-	Video
Park, Spurr, et al [39]	4.5 degree 10.3 degree 3.8 degree	Fully convolutional (Hourglass) and regressive (DenseNet) architecture	MPIIGaze, EYEDIAP, Columbia	150x90	Single Eye image
Park, De Mello, et al. [56]	3.18 degrees 3.42 degrees	Disentangling Transforming Encoder-Decoder (DT-ED)	GazeCapture MPIIGaze	-	Eye image
Zhang, Sugano, Fritz, et al. [25]	10.8 degrees	Own CNN architecture-GazeNet	Own Dataset-MPIIGaze	60x36	Eye image
Zhang, Sugano, Fritz, et al. [42]	4.8 degrees 6 degrees	AlexNet	MPIIGaze, Eyediap	448x448	Full Face
Chen and Shi [57]	4.8 degrees	Own architecture-DilatedNet	MPIIGaze	Eye-64x96 face-96x96	Eye and Face
Fischer et al. [31]	7.7 degrees	Own architecture	Own dataset-RT-GENE	-	Face image
Cheng, Lu, et al. [58]	5 degrees	Own architecture-ARE-Net	Modified MPIIGaze	36x60	Eye Image
Cheng, Huang, et al. [49]	4.1 degrees 5.3 degrees	Own architecture- CA-Net	MPIIGaze Eyediap	Eye-36x60 Face- 224x224x3	Eye and Face Image
Zhang, Sugano, Bulling, et al. [59]	6.6,4.5,3.3 degrees	Own architecture	Eyediap, MPIIGaze GazeCapture	224x224	Face Image
Z. Wang et al. [53]	1.79 degrees	RESNET-34	Own dataset	224x224	Face Image
Deng and Zhu [32]	4.3 degrees	AlexNet	Own Dataset	224x224	Full face and One eye
Palmero et al. [50]	5.1,6.2 degrees	VGG-16	Eyediap	224x224 120x48	Full face two eyes
Jyoti and Dhall [43]	2.22 degree 2.08 cm	Own CNN architecture	Columbia Eye Gaze TabletGaze	-	Face Image
Dubey et al. [33]	91.5%	Own architecture-Ize-Net	Own dataset	128x128x3	Full Face collected from YouTube creative common section
Zhou et al. [60]	4.18,5.84 degrees	Own Architecture	MPII Gaze Eyediap	233x224	Face image, Eye image
Mahanama et al. [61]	2.84,10.04 degrees	Own architecture-Gaze-net	MPII Gaze Columbia Eye Gaze	36x36x1	Eye Images
Lemley et al. [40]	4.918 degrees	Own architecture	MPII Gaze	60x36	Eye Images
Yu et al. [41]	5.7,5.4 degrees	Own CNN architecture	UTMultiview Eyediap	36x60	Eye Images
Lian et al. [36]	4.8 degrees	Own architecture	Own dataset	224x224	Face image
Chong et al. [62]	6.4 degrees	Own architecture	GazeFollow+ Eyediap+ SynHead	227x227	Entire image, face image
Zhang, Huang, Sugano, et al. [34]	Mobile-2.3 Desktop-3.5 Tablet-2.8	Own architecture	Own Dataset	448x448	Face Image
Liu et al. [63]	3,3.8,3.77 degrees	Own architecture	Eyediap, MPIIGaze, UT-Multiview	48x72x3	Eye images
Park, Aksan, et al. [35]	2.49 degrees	Own architecture- GazeRefineNet	Own dataset-EVE	128x128	Both eyes images
Zhang, Sugano, Fritz et al. [64]	13.9 degrees	Own architectur	Own dataset-MPIIGaze	36x60	Eye image

close to real-world scenarios [53]. Deng et al. used two TV screens (60 inches, screen size: 1345 mm * 780 mm) for target presentation and array of 12 cameras for presenting data.
Lian et al. used an Intel RealSense SR300 as RGBD camera to capture data using an Apple iMac machine as display. Depth images were used to provide head pose and 3D eye position information[36]. Zhang et al. used five different devices (mobile phone, tablet, laptop, desktop, smart tv) to capture data[34]. Built-in cameras of tablets, mobile devices and laptop was used. Logitech C910 and Logitech 930e camera was mounted on desktop and smart TV respectively to gather data.
Li et al. used an infrared camera to collect eye data since infrared camera is insensitive to external light changes[51]. Kassner et al. used an eye tracking glasses equipped with a scene camera and one infrared spectrum eye camera for dark pupil detection[54]. Fischer at el. implemented eight motion capture cameras, one RGBD camera and a mobile eye tracking glasses to capture data [31].
The computer is the most common platform for gaze estimation. The cameras are usually installed below/above the computer screen[39], [48], [55]. Some works focus on using deeper neural networks or extra modules[48], [55] to improve gaze performance, while the other works make use of custom devices for gaze estimation, such as multi-cameras and RGBD cameras[31], [36]. Table 4 provides a summary of various gaze estimation techniques for desktops only.
Head-mounted gaze trackers are portable devices with applications ranging from computer input, gaming controls, interactions in virtual environments, augmented reality, and neuro/psychological research. The general setup includes two cameras; one camera pointing at the wearer's eye to detect the pupil; and the scene camera capturing the user's point

of gaze, with sometimes additional components like hot mirrors and NIR light sources. Head-mounted gaze trackers have been implemented as cost-efficient, attachment-free, lightweight devices with simple hardware and software. Also, they are known to provide high-accuracy gaze information in unconstrained settings. Kassner et al. used eye tracking glasses with two cameras: Eye camera and a scene camera[54]. In this, first eye images are converted into a grayscale image, and the initial region of interest is generated. Then ellipse fitting is done to locate the darkest pupil in the IR illuminated eye camera image. Gaze mapping is then done using a transfer function consisting of two bivariate polynomials of adjustable degrees.
Tablets and Smartphones provide a unique paradigm for gaze tracking applications. Gaze tracking on handheld devices is done using the device front camera, one or more IR light sources and various computer vision algorithm.

4 Conclusion

Over the last few decades, eye gaze estimation has received quite a lot of interest from several industrial, academic, and other areas. In this paper, a detailed study of gaze estimation methods is discussed to highlight diversity in various aspects such as gaze estimation basics, feature extraction, architecture implemented to estimate gaze, calibration, datasets, and performance measures implemented in various works.
Lack of homogeneity can be observed in performance evaluation among several works. Some works estimated accuracy in percentage; others have measured accuracy in terms of distance or degrees. This variation makes inter-comparisons between different works improbable. Even though CNN-based deep learning architectures are very effective in estimating gaze, these methods also have some limitations that can become the basis of future works. CNNs are very time-consuming and computationally very expensive. Future research can focus on developing hardware-friendly, computationally inexpensive architectures that do not require any external GPUs or multi-core CPUs for smooth implementation.

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[3] Zhihua Zhou. Machine learning beijing, 2016.

[4] Tom Mitchell, Bruce Buchanan, Gerald DeJong, Thomas Dietterich, Paul Rosenbloom, and Alex Waibel. Machine learning. *Annual review of computer science*, 4(1):417–433, 1990.

[5] Yunpeng Chen. School of computing. *University of Ulster at Jordanstown, Jordanstown, United Kingdom*, 2018.

[6] Smirti Dwibedi, Medha Pujari, and Weiqing Sun. A comparative study on contemporary intrusion detection datasets for machine learning research. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6. IEEE, 2020.

[7] Randall Davis and Jonathan J King. The origin of rule-based systems in ai. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*, 1984.

[8] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[10] Nischal Sanil, V Rakesh, Rishab Mallapur, Mohammed Riyaz Ahmed, et al. Deep learning techniques for obstacle detection and avoidance in driverless cars. In *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*, pages 1–4. IEEE, 2020.

[11] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadlamani Ravi, and Alan Peters. A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems*, 194:105596, 2020.

[12] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[13] David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34, 1995.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[16] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[21] Ruilin Zhao, Yanbing Xue, Jing Cai, and Zan Gao. Parsing human image by fusing semantic and spatial features: A deep learning approach. *Information Processing & Management*, 57(6):102306, 2020.

[22] Abolfazl Abdollahi, Biswajeet Pradhan, Nagesh Shukla, Subrata Chakraborty, and Abdullah Alamri. Multi-object segmentation in complex urban scenes from high-resolution remote sensing data. *Remote Sensing*, 13(18):3710, 2021.

[23] Jun Hee Kim, Haeyun Lee, Seonghwan J Hong, Sewoong Kim, Juhum Park, Jae Youn Hwang, and Jihwan P Choi. Objects segmentation from high-resolution aerial images using u-net with pyramid pooling layers. *IEEE Geoscience and Remote Sensing Letters*, 16(1):115–119, 2018.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[25] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.

[26] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, pages 255–258, 2014.

[27] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 271–280, 2013.

[28] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[29] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5):445–461, 2017.

[30] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6912–6921, 2019.

[31] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European conference on computer vision (ECCV)*, pages 334–352, 2018.

[32] Wangjiang Zhu and Haoping Deng. Monocular free-head 3d gaze tracking with deep learning and geometry constraints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3143–3152, 2017.

[33] Neeru Dubey, Shreya Ghosh, and Abhinav Dhall. Unsupervised learning of eye gaze representation from the web. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[34] Xucong Zhang, Michael Xuelin Huang, Yusuke Sugano, and Andreas Bulling. Training person-specific gaze estimators from user interactions with multiple devices. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[35] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. Towards end-to-end video-based eye-tracking. In *European conference on computer vision*, pages 747–763. Springer, 2020.

[36] Dongze Lian, Ziheng Zhang, Weixin Luo, Lina Hu, Minye Wu, Zechao Li, Jingyi Yu, and Shenghua Gao. Rgbd based gaze estimation via multi-task cnn. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 2488–2495, 2019.

[37] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the symposium on eye tracking research and applications*, pages 207–210, 2014.

[38] Yifan Xia, Baosheng Liang, Zhaotong Li, and Song Gao. Gaze estimation using neural network and logistic regression. *The Computer Journal*, 65(8):2034–2043, 2022.

[39] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–738, 2018.

[40] Joseph Lemley, Anuradha Kar, Alexandru Drimbarean, and Peter Corcoran. Convolutional neural network implementation for eye-gaze estimation on low-quality consumer imaging systems. *IEEE Transactions on Consumer Electronics*, 65(2):179–187, 2019.

[41] Yu Yu, Gang Liu, and Jean-Marc Odobez. Deep multitask gaze estimation with a constrained landmark-gaze model. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[42] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 51–60, 2017.

[43] Shreyank Jyoti and Abhinav Dhall. Automatic eye gaze estimation using geometric & texture-based networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2474–2479. IEEE, 2018.

[44] Zhuoqing Chang, J Matias Di Martino, Qiang Qiu, Steven Espinosa, and Guillermo Sapiro. Salgaze: Personalizing gaze estimation using visual saliency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[45] Junfeng He, Khoi Pham, Nachiappan Valliappan, Pingmei Xu, Chase Roberts, Dmitry Lagun, and Vidhya Navalpakkam. On-device few-shot personalization for real-time gaze estimation. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[46] Tianchu Guo, Yongchao Liu, Hui Zhang, Xiabing Liu, Youngjun Kwak, Byung In Yoo, Jae-Joon Han, and Changkyu Choi. A generalized and robust method towards practical gaze estimation on smart phone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[47] Matthew Kim, Owen Wang, and Natalie Ng. Convolutional neural network architectures for gaze estimation on mobile devices. *Standford, CA: Standford University.[Google Scholar]*, 2016.

[48] Anjith George and Aurobinda Routray. Real-time eye gaze direction classification using convolutional neural network. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2016.

[49] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10623–10630, 2020.

[50] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *arXiv preprint arXiv:1805.03064*, 2018.

[51] Bin Li, Hong Fu, Desheng Wen, and WaiLun Lo. Etracker: A mobile gaze-tracking system with near-eye display based on a combined gaze-tracking algorithm. *Sensors*, 18(5):1626, 2018.

[52] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017.

[53] Zhecan Wang, Jian Zhao, Cheng Lu, Fan Yang, Han Huang, Yandong Guo, et al. Learning to detect head movement in unconstrained remote gaze estimation in the wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3443–3452, 2020.

[54] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*, pages 1151–1160, 2014.

[55] Sheng-Wen Shih, Yu-Te Wu, and Jin Liu. A calibration-free gaze tracking technique. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 201–204. IEEE, 2000.

[56] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9368–9377, 2019.

[57] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018.

[58] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 100–115, 2018.

[59] Xucong Zhang, Yusuke Sugano, Andreas Bulling, and Otmar Hilliges. Learning-based region selection for end-to-end gaze estimation. In *BMVC*, 2020.

[60] Xiaolong Zhou, Jianing Lin, Jiaqi Jiang, and Shengyong Chen. Learning a 3d gaze estimator with improved itracker combined with bidirectional lstm. In *2019 IEEE international conference on Multimedia and expo (ICME)*, pages 850–855. IEEE, 2019.

[61] Bhanuka Mahanama, Yasith Jayawardana, and Sampath Jayarathna. Gaze-net: Appearance-based gaze estimation using capsule networks. In *Proceedings of the 11th augmented human international conference*, pages 1–4, 2020.

[62] Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James M Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 383–398, 2018.

[63] Gang Liu, Yu Yu, Kenneth A Funes Mora, and Jean-Marc Odobez. A differential approach for gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):1092–1099, 2019.

[64] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.