National College of Ireland

Innovative Study on Popular Approaches Used in Gaze Prediction

MSc Research Project
Data Analytics

Sammam Sohail
Student ID: 23256800

School of Computing
National College of Ireland

Supervisor: Anu Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sammam Sohail……..……

Student ID: 23256800……..…….

Programme: MSc Data Analytics…… **Year:** 2024……

Module: MSc Research Project……..……

Supervisor: Anu Sahni……..…….

Submission Due Date: 12/12/24……..……

Project Title: Innovative Study on Popular Approaches Used in Gaze Prediction…..……

Word Count: 9564…… **Page Count**…25……..……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Sammam Sohail……

Date: ……11/12/24……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Innovative Study on Popular Approaches Used in Gaze Prediction

Sammam Sohail
23256800

Abstract

The rapid advancement in technology today has given significant importance to gaze prediction. Ranging from virtual reality to monitoring humans in a specific environment, gaze prediction plays a vital contributor in these fields. This paper focuses on implementing a system that can monitor human gaze direction using the traditional techniques of computer vision and the more advanced convolutional neural networks. The traditional technique implemented in this paper relies on the famous HOG classifier for extracting human eyes from face images and a modular approach for facial feature extraction using convolutional neural networks for gaze prediction. We use the Columbia Gaze dataset, which is a popular dataset in the field of gaze prediction, for training and evaluating the systems developed for this paper. The aim of this study is to develop and exploit the limitations of each technique that predicts the human gaze. The paper, at the end, discusses various analysis techniques, comprising visualisations and evaluation metrics such as precision, recall, f1-score and accuracy. To contribute further, a recommendation is made by evaluating the metrics between the employed techniques for their use in different gaze prediction environments.

1 Introduction

1.1 Background

Gaze prediction is a field of computer vision which refers to the identification and monitoring of the direction in which a person looks. The ability of computers to detect and track eye movement has given rise to superior methods of analysing human behaviour and the way humans interact with machines for example, vehicles equipped with driver monitoring use gaze detection to monitor driver attention, online exams and recruitment test portals use proctoring technology that use gaze detection to check if a candidate is using multiple displays.

Tracking the movement of eyes can help assist in determining the attention, focus, and intentions of individuals, which makes this beneficial across many industries. Many fields, such as advanced driver assistance systems (ADAS), behaviour analysis, simulators, the gaming industry, and assistive technologies for the disabled, use this technology as a core feature. In a study performed by (Ryu et al., 2024), gaze monitoring was combined with behaviour analysis to evaluate human reaction in road accident situations, the results of the implemented GCN model showed a notable contribution of eye behaviour in the overall analysis, which proves the influence of gaze prediction in human monitoring. Furthermore,

extensive research is being conducted to improve the interaction between humans and robots. For this purpose, (Samaniego et al., 2024) explored the gaze prediction module in pepper robots, emphasising that it is essential for the robots to understand human expressions to improve social interaction.

1.2 Motivation

Throughout the years, the evolution in machine learning has led to the development of various approaches used to detect human gaze emerging from simple machine learning techniques to the more advanced convolutional neural networks. However, despite the large contributions made to this field, effective gaze prediction proposes new challenges across various real-world environments and systems due to lighting conditions, facial features, head pose and limitations in hardware. These challenges are further highlighted by (Chandel et al., 2024) where the authors insist on addressing these challenges for an efficient integration of gaze prediction systems in IoT devices for paediatric healthcare. Thus, due to these challenges developers and researchers spend substantial time researching what approach to follow in their use case.

This research investigates the two state-of-the-art technologies used for human gaze prediction. The first technique utilises the OpenCV and Dlib library, which incorporates simple machine learning models such as the Histogram of Oriented Gradients (HOG) for extracting eye regions from face images. On the other hand, the modular convolutional neural network system developed for this study uses a pipelined approach to detect face, eye regions, and finally predict the gaze direction from the given image. The goal is to implement robust systems and investigate the competence of each system to provide future researchers and developers recommendations and starting points for their applications through various analysis techniques.

1.3 Research Question

Which gaze prediction technique should be used in a particular gaze detection application?

(J. Li et al., 2022) conducted a comparative study on the different techniques used for gaze prediction, such as template-matching, machine learning models, and neural networks. For future work, the researchers focused on providing a detailed explanation and comparison of the system architecture of the individual techniques. It is not only essential to link and evaluate the results with the mechanism of the system but also necessary to connect the outcomes of the comparison to the specific requirement of gaze prediction applications. Further, their study utilised a self-collected dataset, which can sometimes be inefficient for future evaluation and simulating real-world scenarios such as head-pose angles.

1.4 Research Objective

The objective of this research is to implement gaze prediction systems using OpenCV and CNNs to reveal the trade-offs and advantages of each technique such as the performance in

resource-abundant and resource-scarce environments. The list below highlights the steps carried out for developing the gaze prediction systems for this research:

- Develop and implement gaze prediction using OpenCV and Dlib libraries.
- Design and train CNN-based models.
- Evaluate the systems developed for this research.
- Identify and evaluate models for different environments.

2 Related Work

Previous research in gaze prediction has led to many advancements in the technology used for gaze prediction systems. The recent work utilises state-of-the-art tools, particularly OpenCV and CNNs, to build robust eye-tracking systems. Even though being a legacy technology, OpenCV provides quality results with efficiency due to its wide adoption and accessibility in hardware-constrained environments. As machine learning has seen huge growth over the years and computational resources are more readily available to the public, new approaches such as CNNs are being adopted in gaze prediction systems offering promising solutions in complex execution environments.

2.1 Traditional Techniques Used in Gaze Prediction

This approach for predicting gaze has been used extensively due to the feasibility of running on simple hardware and producing effective results. The systems designed on these techniques rely on a pipelined approach by first detecting the face region from an image, then identifying the eye regions, and finally predicting the gaze direction. These steps remain similar across many of the previous research however, the tool or algorithm used varies.

(Viswanatha et al., 2023) produced a low-cost eye movement-controlled wheelchair that utilised Haar Cascade for face extraction and template matching technique for eye region extraction. The research elaborates on the Haar Cascade algorithm, which works by using square functions computed through the principles of integral image. The output of the Haar cascade algorithm produces a huge set of results, which is narrowed down using the Adaboost technique. To detect the gaze direction, a 68-point facial landmark template is used to detect the eye regions, these cropped regions are then binarized followed by calculating the ratio of dark regions in an eye to identify the location of the pupil, therefore determining the gaze direction. The system developed in this research produced robust results but with a high inference time compared to other algorithms due to the use of ensemble learning. (Kathpal et al., 2021) used the same template-matching technique to develop a typing application for disabled people utilising eye tracking which also demonstrated robust results with lesser inference time compared to the system produced using the Haar Cascade classifier.

(Maralappanavar et al., 2016) established a system in their study to monitor driver distraction by utilising gaze monitoring. They implemented the Viola-Jones algorithm, which is used for object detection using an algorithm pipeline and training data to detect face regions from an image. It is influential to note here that Haar Cascade is an application of the Viola-Jones algorithm and is narrowed down to pre-trained models that are ready to use. The

face region produced by the Viola-Jones algorithm is then converted to a YCbCr colour space, and edge detection is used to produce a rough estimate of the eye regions. Finally, these eye regions are used to calculate the distance of the pupil from the eye corners. The results of the study showed an accuracy of 75% but struggled to maintain this accuracy in low-light environments. From this study, it can be seen that the Viola-Jones technique can lead to limited accuracy due to adaptability problems in gaze prediction environments.

The feature of low processing requirements in traditional techniques, hybrid systems have been introduced by researchers, which harvest computer vision algorithms for some of the initial steps of gaze prediction. (Cao, 2023) utilised a hybrid approach in which face prediction was performed using the HOG classifier and the later steps, such as eye region detection, were performed using SVM. HOG classifier works by encoding of intensity gradients and edge directions for identifying areas of interest. It is a popular tool in less resource-available object detection systems where the shape of the object is a distinct feature. The results of another approach by (K R et al., 2023) showed that the use of the HOG classifier for face detection in a hybrid approach produced less error during training and testing, improving the overall results of the system.

The work done by (Mounica et al., 2019) brings a hybrid approach in which the face and eye region localisation is performed by the Haar cascade classifier, and the gaze direction is predicted by a neural network. To simplify the complex computation, the system only used the left eye to perform gaze detection. This step ultimately reduced the overhead of training a sophisticated model. However, this step hides the left eye when a person is looking to their left, causing the model to fail. Further, testing phase of the neural network in this research was capable of training on new images, which can cause the size of the neural network model to increase and, therefore, reduce inference speed. Another system made by (Yuli Cristanti et al., 2017) incorporated gaze prediction for developing assistive pointer control for people with physical disabilities. The system used the same Haar cascade classifier for face detection, and for the next step, calculations comprising of face width and height were used to perform eye region localisation for the left eye. It is crucial to note here that face dimensions vary from person to person and can lead to errors in systems when used across different test subjects. This proposed system then performed iris segmentation using inverse binary threshold and morphology to locate the iris in the eye image. To identify the gaze direction, the x-axis movement of the iris was tracked where an x value greater than the initial value was found the system outputs a right gaze and vice-versa to output a left gaze. This system showed some restrictions, such as the limitation of working at a maximum eye and camera distance of 100cm and poor performance in low light conditions.

Unmanned aerial vehicles (UAVs) require the training of individuals before they can be deployed. To provide a simpler solution to UAV control, (Zhu et al., 2021) utilised eye tracking to manoeuvre UAVs. As a first step for eye tracking, a template matching technique is utilised by the authors for eye region extraction. To track the pupil, the authors use IR-LEDs with a camera to track the pupil movement in video frames. The use of IR-LEDs significantly reduces the preprocessing as face images captured in IR light significantly highlight the pupil. This technique provides a much simpler solution to gaze tracking but requires special hardware, controlled lighting, and relies on video frames to track the x and y-axis movement of the pupil in subsequent frames. Further, the more hardware is used in a

system, the chances of equipment failure significantly increase, which can lead to catastrophic results if proper failsafe measures are not used, such as in the case of UAV control.

2.2 Advanced Techniques Used in Gaze Prediction

CNNs are one of the cornerstones in the field of computer vision. Many tasks that were computationally challenging using machine learning have been made possible by CNNs due to their robust and accurate feature extraction capabilities, unlike traditional machine learning techniques, which require manual feature extraction. CNNs have the advantage of capturing spatial and semantic patterns in data, making them robust to different head poses, and illumination problems, ensuring reliability in real-world and diverse scenarios. The review below explores some of the architectures used by previous researchers to develop the CNN models.

The system proposed by (Ewaisha et al., 2020) implemented a convolutional neural network which predicts driver gaze direction and head pose detection for automotive safety. The authors of the paper approached the task as a regression problem in which their model predicted the gaze and head pose angles at the same time. This use of this technique helped the model to better understand the relationship between the gaze and head pose angles, which can be a difficult task to perform when two separate models are used for each task. The dataset they utilised was the Columbia Gaze dataset, along with dataset augmentation, which increased the size of the dataset to 45000 images. For the architecture, eight convolutional layers were used, followed by dense layers with weights initialised using the VGG face descriptor model. The final model achieved an accuracy of 78.2% in testing. It is realised here that using weights of a pre-trained model for further training can cause the model to overfit. Furthermore, using a low number of layers can obstruct the model's ability to learn and differentiate between two different tasks.

(Herashchenko & Farkaš, 2023) developed a modular system for gaze prediction comprising artificial neural networks. The authors used transfer learning in their architecture for the face (RetinaFace) and head pose prediction (6DRepNet), along with custom CNN layers for estimating gaze and head pose angles. A synthetic dataset was made for the research and combined with the Columbia Gaze dataset for training and testing the system. Pipelining these individual tasks has the advantage of training and evaluating every module separately, therefore increasing the reliability of the system. The results of the system prove this factor, which deliver an MAE of less than 2 degrees in both head and gaze angle predictions.

(Lemley et al., 2019) came up with a low hardware utilising gaze detection system based on a CNN model that took cropped eye images as input and estimated the gaze angle from both eyes. The use of both eyes for predictions is important as head pose can sometimes cause the portion of an eye in the image to disappear, resulting in the failure of the model to locate the eye. The authors utilise the MPII Gaze dataset and highlight that small and high camera distance images in the dataset have a poor impact on the model's performance. Keeping a note of this, it is essential to perform augmentation on the dataset. The CNN architecture was made up of 3x3 stacked convolutional layers, which reported a mean error of

3.65 degrees. The results of this work show that careful data preprocessing is an important step when there is a dataset quality issue.

(Ansari et al., 2021) highlight in their paper that gaze prediction requires special hardware and proper lighting environments. To solve this issue, they designed a lightweight CNN model with no requirement for special hardware. For training this model, the researchers utilised a simple webcam to collect the images for different gaze angles however, the use of a low-quality webcam resulted in low-quality eye images. The first model out of three utilised a single eye for gaze detection, the second used full-face images, and the third included both eye images for predicting gaze directions. It was reported that the model that utilised the left eye images was able to produce the highest accuracy of 88% on front-face images. The review of this study shows that the system was not tested on images with head pose, and for the system to be reliable, it should be tested on all real-world scenarios. The findings made by the researchers show that even though the model was not able to compete with the best systems, its simplicity makes it useful for human-computer interactions.

(Prince et al., 2020) put stress on the use of eye tracking in the bio-medical field for the diagnosis of many diseases, such as Alzheimer's. They further highlight the issues related to system design, for example, the requirement for a high-quality camera, the need to regularly update the model, and the availability of a quality dataset. The authors used the Eye Chimera dataset and experimented with different techniques for face detection, such as HOG classifier, neural networks and template matching using facial landmarks. Finally, the system developed with CNN, SVM, and Electrooculography reflected the best results in classifying eight gaze directions. To improve the efficiency of the model, the authors recommend increasing the size of the dataset and incorporating RNN for real-time prediction purposes. The findings and challenges faced by the authors highlight the need to address the problems associated with the dataset used in a study.

The paper by (Hu et al., 2020) introduced a CNN model named DGaze for predicting gaze directions in different environments. The authors collect eye tracking data from 43 different subjects in 5 different scenes, including both indoor and outdoor conditions. The statistical analysis of the dataset revealed that the gaze direction, head pose velocities, position of objects the individuals looked at, and saliency (attention regions) of visual features were correlated with each other. Taking note of this statistical analysis, a CNN model was developed, which took encoded sequences of head velocity, position of objects, and gaze position as input, along with saliency maps extracted from the images as a separate input. It can be inferred here that producing a high number of input variables can cause a pre-processing overhead, ultimately reducing the efficiency of the model. The results produced by the DGaze model outperformed the previous SGaze model, showing a 22% improvement in dynamic and a 9.5% improvement in static scenes. These results produced by the model were further evaluated using an ablation study where the contribution of each of the inputs to the model was verified. The results of the DGaze model show a huge improvement, but the model is not evaluated in real-world scenarios and there is a trade-off of resource and preprocessing overhead.

2.3 Evaluation of Gaze Prediction Techniques

A review of previous research highlights the issue of different techniques used by researchers for evaluating gaze prediction systems. (Kar & Corcoran, 2017) emphasised the use of standard evaluation metrics, i.e. precision, recall, F1-Score, etc, to cross-evaluate different gaze prediction systems. The same authors (Kar & Corcoran, 2018) presented a software tool in their other study to mitigate the issue of different evaluation metrics. They utilised powerful evaluation metrics and visualisations that provide quantitative and statistical results to benchmark different gaze prediction systems. In another research, (J. Li et al., 2022) and (Xiang et al., 2021) perform experiments on CNN-based gaze prediction systems, including the same metrics highlighting the run-time performance and insights for each of the systems.
The review of the evaluation techniques narrows down the metrics that can be used in this study for the individual and cross-evaluation of gaze prediction systems that are developed for this study. However, these analyses studies lack the absence of architecture specifications for the gaze prediction system, which is essential to debug performance issues, provide insights into model behaviour in different execution environments and optimise the resource usage of the system.

Reviewing the existing work done on gaze prediction unveils the methods and approaches that need to be followed for developing efficient gaze prediction systems. Understanding the core features and working of each of the techniques can help in addressing and handling any issues encountered during the development of each approach. Further, this review identifies the downsides of each technique, which can help in producing a better implementation of the previous work.

3 Research Methodology

To explore and evaluate the gaze prediction systems proposed for this study, it is essential to follow a certain path and build up on the methods used by previous research for the development of gaze prediction systems.

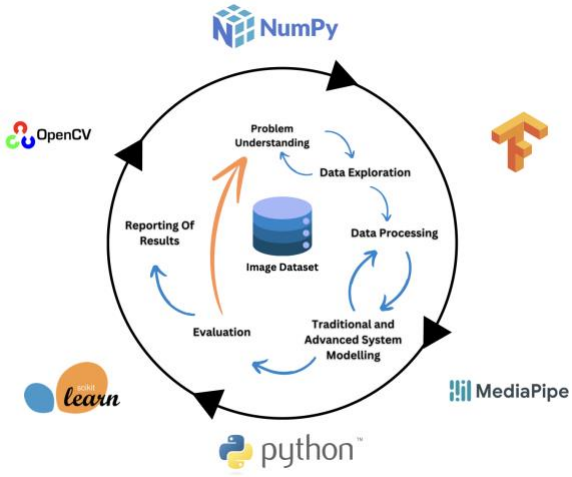

Figure 1: Methodology plan and tools

To reach the objectives of this research, a modified version of the CRISP-DM process shown in Figure 1, is used to align the development of different phases of this study. The

project plan includes the steps of problem understanding, data exploration and preprocessing, system modelling, evaluation, and reporting of results, each section is explained in the following individual sub-sections.

3.1 Problem Understanding

Gaze detection is a crucial field in computer vision, with its usage applications spanning from high-tech devices to medical diagnosis usage. The two most popular methods used in predicting gaze direction include tools such as OpenCV and CNNs. The problem addressed in this research is to identify the use case of each technique by evaluating the trade-offs and advantages of each technique. The goal of this study is to analyse for what scenario a particular technique should be used. To achieve this, the study plans to build separate gaze prediction systems based on OpenCV and CNN and analyse the performance and results to get a better view of what technique can be used in a particular gaze prediction environment.

3.2 Dataset Exploration and Preprocessing

To train and evaluate the systems in this research, it is essential to gather data that is most used by previous research for conducting a thorough analysis. From the literature review, it was seen that public datasets and self-collected datasets are used extensively. It is essential here to highlight that self-collected datasets may give rise to several problems, such as bias in the model, overhead of arranging resources and test subjects. Further, to cross-evaluate the developed model with previously developed models many conditions need to be kept the same in which dataset plays a key factor.

The Columbia Gaze dataset, prepared by (Smith et al., 2013), is a widely used dataset in the field of gaze prediction and has been used to develop and evaluate many of the gaze prediction algorithms, for this study, the Columbia Gaze dataset is used due to a variety of features. The dataset contains 58880 face images of 56 people (32 male and 24 female) with different ethnicities to ensure diverse facial and eye features. Each person in the dataset is photographed under standard lighting conditions (to minimise variability) with five vertical head poses, seven horizontal gaze directions, and three vertical gaze directions. The key features of this dataset include:

- High-resolution images (5184 x 3456 pixels).
- Diverse head poses simulating real-world scenarios.
- 7 gaze directions, enabling CNN models to learn variations.
- Test subjects wearing spectacles
- Labelled gaze directions and head poses in the form of angles

Even though this dataset provides controlled variations, there are some challenges associated with it. The dataset has an absence of dynamic background, limiting the complexity of environments, such as those seen in driver and online exam monitoring scenarios. Further, the size of the dataset is small compared to other datasets, such as the MPII gaze dataset, which contains millions of images (Y. Li & Li, 2024). Another issue

associated with this dataset is that there is a need to develop the bounding box data for the eye regions in the images. Much of the previous research uses cropped eye regions as input for the gaze detection systems, and the Columbia Gaze dataset does not provide this information directly, requiring researchers to use their techniques, such as manual and automatic tools, for producing cropped eye regions. To cater to such issues for this study, techniques such as face region cropping, removal of extreme head poses, and grayscale conversion are used. The later sections of this study will elaborate on the tools and methods used for the pre-processing before using this dataset in the gaze prediction systems for this study.

3.3 Traditional and Advanced System Modelling

For this research, the gaze prediction systems are categorised into traditional and advanced approaches. These systems differ from each other significantly in terms of methodology, computational resource requirements and the adaptability to handle complex scenarios. These two techniques are designed and evaluated separately for this research to provide a detailed explanation of the strengths and weaknesses of each technique.

The traditional gaze prediction designed for this study relies on classical algorithms of computer vision, particularly HOG (Histogram of Oriented Gradients) and template matching technique, along with rule-based methods to detect the gaze direction. The previous research by (Miah et al., 2022) shows that these types of systems are used popularly in environments with less computational resources and real-time inference. The advanced gaze prediction system developed for this study uses transfer learning and convolutional neural networks. This study experiments with a direct approach and a pipelined CNN approach for a deeper investigation of the advanced system. The direct approach uses CNNs to directly estimate the gaze direction from an input image whereas the pipelined approach uses face detection, eye region detection and gaze detection module. Previous research shows that advanced techniques are more suitable in handling complex scenarios such as head-poses and lighting issues but demand more computational resources compared to the traditional algorithms thus, giving an advantage to traditional approaches in the areas where computing power is low.

As computer hardware has improved over the years, more processing power is available on smaller hardware devices, such as the latest Raspberry Pi 5, which provides enough computing power to run lightweight neural networks (Al Hamdi et al., 2023). To cater for the issue of the heavy resource requirement of CNNs, pre-trained models that provide a balance in performance and efficiency, such as MobileNet is used in this research.

3.4 Evaluation

To achieve the goal of this study, it is essential to follow the techniques that were used by previous research to evaluate the systems. (Kar & Corcoran, 2017) highlighted a significant issue with gaze prediction systems, which is the lack of standard tools used for the evaluation of systems. In their next study, they prepare a software which solves this issue by using numerical and visual techniques to gauge the performance of the gaze prediction systems. This technique is also used by (J. Li et al., 2022) where the same set of metrics were used to

evaluate the performance of gaze prediction systems. These metrics include recall, precision, accuracy, f1-score, and different visualisations which will be used for this study as well.

4 Design Specification

The system design for this research follows a systematic approach for developing the CNN model (advanced gaze prediction) and OpenCV model (traditional gaze prediction) to answer the research question of this study.

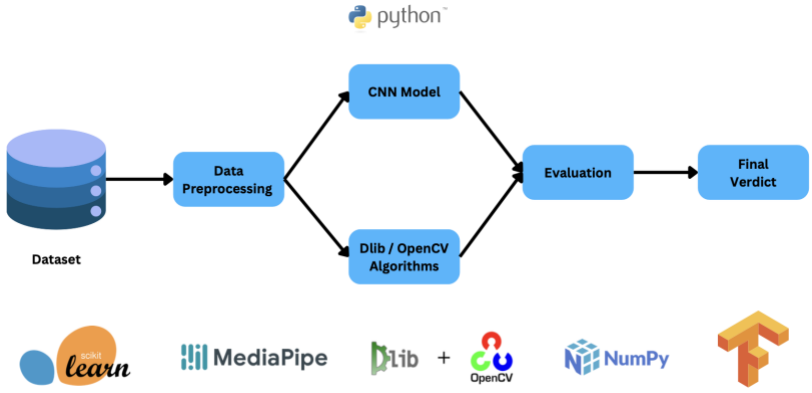

Figure 2, Design Specification Diagram

Figure 2 highlights the design specification diagram. The first step was to load the Columbia Gaze dataset in the Visual Studio Code IDE to perform the necessary preprocessing to prepare the data for the CNN and OpenCV systems. Several steps, such as image resizing and cropping were performed to prepare the data for training and testing of the systems.

The CNN models represent the deep learning-based systems developed for this study. The CNN models process the data prepared in the previous step for identifying complex facial patterns for predicting the gaze direction. The OpenCV model represents the traditional algorithm-based gaze prediction system developed for this study. The traditional algorithm incorporates the use of Dlib and OpenCV libraries, providing pre-trained models for facial feature detection and facial template matching which are the classical approaches used in face detection-based applications.

The results produced by the CNN model and OpenCV algorithms are then fed to the evaluation step, where the results are compared and evaluated using various metrics and visualisations mentioned in the previous section. Based on these evaluations, the deployment of these approaches in various environments is discussed for future research references.

5 Implementation

In this study, the gaze prediction systems are categorised into traditional and advanced approaches. The systems differ from each other significantly in terms of methodology, computational resource requirements and the adaptability to handle complex scenarios. To train and test the advanced gaze prediction systems of this study, Kaggle Cloud notebooks are used, and the traditional gaze prediction system is developed and tested on Apple M3 SOC.

The below sections will elaborate on the architecture, tools, and frameworks used to develop each of the systems.

5.1 Dataset Preprocessing

The preprocessing steps implemented in this study needed to be addressed thoroughly to minimise the limitations of the dataset and to prepare it for an efficient implementation of gaze prediction systems. The steps implemented are as follows:

1. **Face Cropping and Eye localization:** This step of image preprocessing was performed using Google's MediaPipe library. This tool provides ready-to-use functions for various computer vision and machine learning tasks with reliability and efficiency. Using this tool, the original images were cropped to keep the face region in the images, and these face photos were then used to create the bounding box coordinates of the left and right eyes.
2. **Removing Extreme head poses**: In the previous step, it was observed that the MediaPipe functions for eye region detection failed to localize eye regions at head pose above -15 and +15 degrees therefore, it was essential to remove such images as this could produce error in the dataset. This step reduced the size of the dataset to 3197 images.
3. **Classification and Encoding of Gaze Directions:** To reduce the processing overhead in the OpenCV implementation and to categorise the gaze directions, the Sklearn library was utilised to convert the gaze angles into three classes with all negative angles categorised as left gaze, all positive angles as right gaze and the 0 angles categorised as centre. In the end, these classes were one-hot encoded to ensure the compatibility of the dataset with the output layer of the CNN model.
4. **Cropping and Resizing:** To meet the requirements of the CNN model, the original images were resized to 244 x 244 pixels using the OpenCV library.
5. **Conversion to Grayscale Images:** The images in the dataset were converted to grayscale using OpenCV's image colour conversion methods to reduce the computation complexity, keep an influence on edges and contrast in the images and minimise the influence of eye colours. However, to follow the input requirement of 3 colour channels for pre-trained CNN models, the grayscale images were expanded to 3 dimensions for the advanced gaze prediction systems using the Numpy library.
6. **Pixel Normalization (For Advanced Modelling):** Keeping in mind the previous research, the pixel values were normalized to values between 0 and 1 range. This step reduces variability caused by lighting conditions.
7. **Annotation Refinement:** To combine the eye-bounding box data and to remove the head-pose directions, a new CSV file was created to store all the labels to ensure simplicity and to remove the overhead of separate label files.
8. **Data Splitting:** For the training and testing sets, 70% of the data was allocated for training and 30% was allocated for testing the systems.

5.2 Traditional Gaze Prediction System

This system relies on classical computer vision algorithms used for detecting facial and eye regions for gaze detection. These techniques are known for their low resource requirement while producing promising results. For this system, the OpenCV and Dlib libraries are utilised as they are popular tools for computer vision tasks, and almost all the algorithms and techniques used by previous research are available in this library.

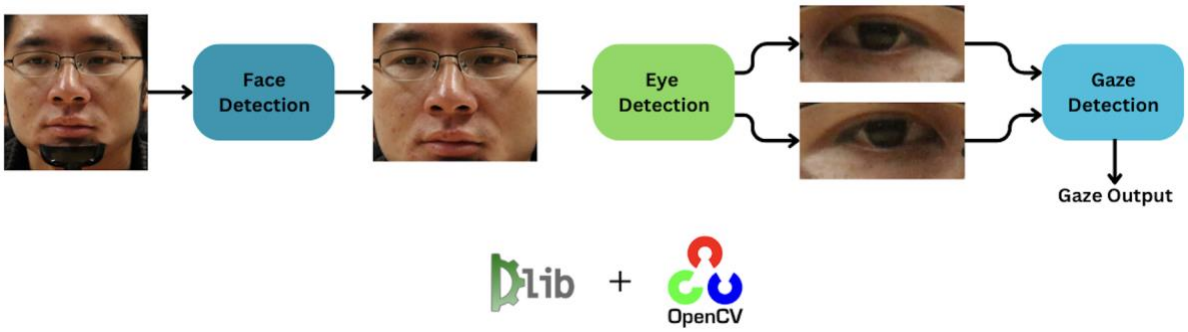

Figure 3: Pipeline for traditional system

Figure 3 outlines the architecture for the traditional gaze prediction system developed for this research, which follows the classical approach used in the previous work. The first stage of the system uses a Histogram of Oriented Gradients (HOG) classifier to detect the face region from the input image. The HOG classifier is a pre-trained SVM classifier that detects areas of interest by capturing edges and textures, making the system more robust to lighting variations, which mitigates a popular issue seen in the previous work. The face detection stage narrows down the area of interest for the next detector stage, therefore allowing the detector to focus on more relevant areas.

The eye detection stage uses a popular technique known as the template matching algorithm to detect individual features of the face, such as nose, eyes, mouth, etc. This algorithm incorporates an ensemble model of regression trees trained on a labelled facial landmarks dataset to output the 68 coordinates of facial landmarks. The landmarks present on points 37 to 48 are the eye regions which are sent to the final stage of the system.

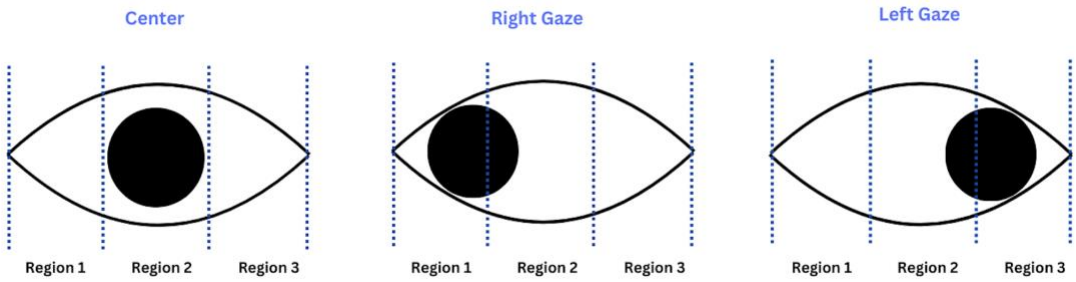

Figure 4, Ratio method for detecting Gaze direction

The gaze detection stage uses the eye landmarks from the previous stage to crop eyes from the face image. These cropped images are then used to predict the gaze direction by utilising the ratio method. This method works by first converting the images to binary, which leaves the area around the pupil as 0 and the area covering the pupil as 1 by utilising a threshold

value between 0 and 255 as a cutoff point. The pixel values below the threshold value are the dark regions making up the pupil therefore, the values are set to 1 and pixel values above the threshold value are the white area around the pupil hence, they are set to 1. Figure 4 highlights the further working of the ratio method, where the eye image is divided into three segments to calculate the position of the pupil. The pixel values in each of the segments are summed, and the region with the highest value is where the pupil is positioned, hence corresponding to the direction where the eye is looking. This method is applied for both eye regions, and a particular gaze direction is returned by the system when both eyes gaze in the same direction, e.g. if both eyes gaze in the right direction, then the output of the system is right, any different directions are considered as undetectable by the system.

5.3 Advanced Gaze Prediction System

This system refers to the use of advanced machine learning models, such as neural networks that can uncover and learn complex data patterns. The previous research particularly, by (Herashchenko & Farkaš, 2023) and (Ewaisha et al., 2020) utilises transfer learning and convolutional neural networks to develop robust gaze prediction systems. The following paragraphs explains the architecture of the two CNN models that are developed using the TensorFlow library for the advanced system of this study.

5.3.1 Direct Approach CNN Model

The architecture for the direct approach model uses the facial feature extraction ability of the VGG-Face model at its core followed by custom CNN and max pooling layers to learn the data patterns for predicting gaze directions. (Ewaisha et al., 2020) used the weights of the VGG-Face model in the first four layers of their CNN model, which emphasises the power of transfer learning in their study to produce quality performance in their gaze prediction solution.

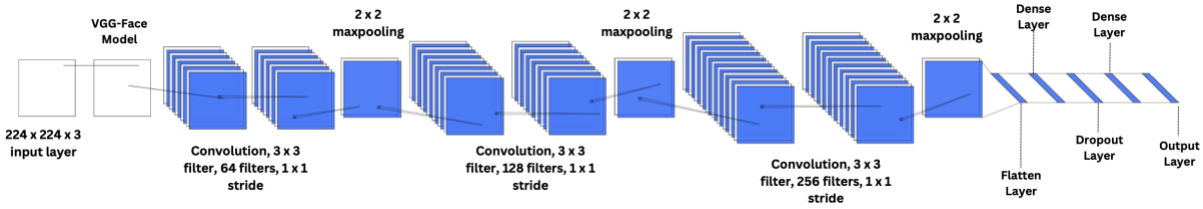

Figure 5, CNN Direct Approach Architecture

This CNN model in Figure 5 follows the same structure used by (Ewaisha et al., 2020), with the VGG-Face model as the base model with an input shape of 224 x 224 x 3. The base model is followed by CNN layers with the ReLU activation function to achieve a detailed feature extraction. A max pooling layer with a 2 x 2 kernel is attached after every 2 CNN layers to reduce spatial dimension while preserving relevant information. The feature maps produced by the feature extraction phase are then flattened, followed by a dense layer with 256 units and another dense layer of 128 units. With a large model, it is essential to keep in mind the problem of overfitting, so a dropout layer is added with a rate of 0.3. Finally, a SoftMax-activated output layer is attached to the model to produce probabilities for the three

gaze directions. In the end, the model is compiled with the Adam optimiser, categorical cross-entropy as the loss function, and accuracy as the evaluation metric.

5.3.2 Pipelined Approach CNN Model

The pipelined CNN model for this study offers a systematic and modular approach to achieving a robust model. (Herashchenko & Farkaš, 2023) used a similar architecture in their gaze prediction study, which reflected several advantages of a modular approach compared to the direct approach. The use of this technique allows the individual CNN modules to be trained, tested and tuned separately, maximising the potential of the whole system.

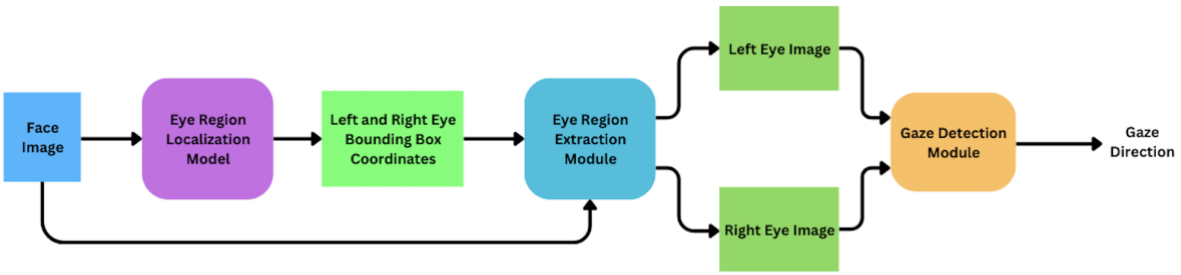

Figure 6, CNN Pipelined Approach Architecture

Figure 6 shows a high-level working of the pipelined approach developed for this study. The first eye region localisation model takes the input image and finds the bounding box coordinates for both eyes. The following module utilises the lambda layer in TensorFlow to crop the eye region using the input face image and coordinates from the previous module. Finally, the Gaze Detection model takes the images of left and right eyes to predict the gaze direction. The next paragraphs further explain the working of the eye localisation, eye region extraction module and gaze detection module.

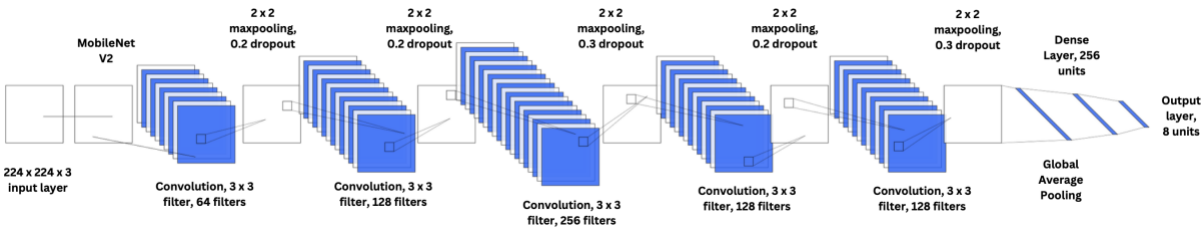

Figure 7, Eye Region Localization Module Architecture

The eye region localisation module shown in Figure 7 uses MobileNetV2 as the base model, followed by custom CNN layers. The input layer shape is kept at 224 x 224 x 3 to meet the requirements of the base model. MobileNetV2 is a pre-trained model trained on the ImageNet dataset which serves as a high-level feature extraction segment in the architecture and thus, all the layers of the pre-trained model are kept frozen to retain its feature extraction capabilities.

The features extracted by the MobileNetV2 model are then refined by custom layers CNN layers with ReLU activation functions. Each convolution layer with a 3 x 3 filter size is followed by a batch normalisation layer for stability, a dropout layer with different rates (depending on the number of filters) to reduce over-fitting, and a max pooling layer with a 2

x 2 kernel is added to reduce the spatial dimensions. The technique of starting with a low number of filters, moving to a high number of filters, and finally reducing the number of filters is inspired by the implementation by (Ewaisha et al., 2020). This technique in building a CNN architecture mimics how human vision works where, at first, low-level features are extracted (edges and textures), and the higher-level features are extracted through successive processing. These high-level features are then summarized by convolution layers with a low number of filters. At the end of convolution layers in the model, a global average pooling layer is added which compresses the spatial dimensions into a single vector followed by dense layers with 256 neurons for a high feature representation and a dropout layer with a rate of 0.3 to maintain regularisation. This dense layer is then attached to an output layer with eight neurons to produce the bounding box coordinates for the two eyes. This model predicts continuous values of the eye coordinates which makes it a regression model therefore, no activation function is used in the output layer. Finally, the model is compiled using an RMSProp optimiser, mean squared error as the loss function, and intersection over union (IoU) as the evaluation metric.

The eye region extraction module present in Figure 6 serves as a sandwich module to provide input to the Gaze detection module. The eye region extraction module utilizes the lambda layer present in the Tensorflow library to crop eye regions using the original image sent as input to pipeline and the bounding box coordinates produced by the eye region localization module. The resultant eye images are then sent to the final gaze detection module for predicting the gaze direction.

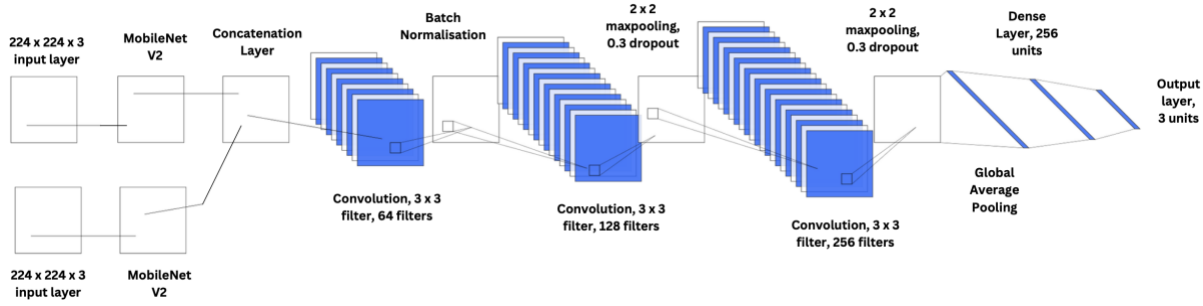

Figure 8, Gaze Detection Module Architecture

The gaze detection module shown in Figure 8 uses a multi-input CNN model for predicting the gaze direction of eye images. In this model, two separate MobilNetV2 models are used to extract visual features from each eye image. The resultant feature vectors are then combined using a concatenation layer, which produces a single feature vector. This combined feature vector is then passed down to the convolution layers in the architecture for further feature extraction. The architecture shown in Figure 7 follows a similar approach and parameters as the eye localisation module. However, this module does not follow the increasing and decreasing number of filter trends, as the input images need to be fully interpreted by the model. In the end, the convolution layers are followed by the global average pooling layer, a dense layer with 256 units, a dropout layer with a rate of 0.4 and an output layer containing three neurons with SoftMax activation function to output the gaze direction (left, center, right). Finally, the model is compiled with the Adam optimiser, categorical cross-entropy as the loss function, and accuracy as the evaluation metric.

6 Evaluation

In this section, different experiments are carried out for each gaze prediction system developed for this research to achieve the best-performing systems. For the evaluation, numerical metrics proposed by previous research, such as recall, precision, accuracy, and f1-score, are used to evaluate the individual systems. Visualisations are also a crucial aspect as they help to visually understand the performance of the models during the different phases of model development, especially in the case of neural networks. Tools such as confusion matrices, accuracy and loss plots, and ROC-AUC curves are used to evaluate class-specific performance.

6.1 Experiments With Traditional System

The traditional system developed for this study uses ready-to-use algorithms therefore, there are very few parameters which can be fine-tuned. Particularly, the binary threshold and centre eye segment ratio parameters are experimented with different values to achieve the optimal system.

Binary Threshold Value	Center Eye Segment Ratio	Accuracy
100	0.25	41%
120	0.25	39%
150	0.25	36%
180	0.25	34%
100	0.2	53%
120	0.2	45%
150	0.2	40%
180	0.2	38%

Figure 9, Experiments with Traditional System

Figure 9 shows the table with different parameter values used to conduct the experiments. From the table, it is evident that a binary threshold value of 100 and a centre eye segment ratio value of 0.2 produced the best result for this technique.

Accuracy	0.53
Precision	0.66
Recall	0.53
F1-Score	0.49

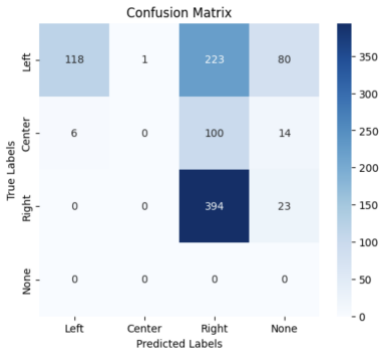

Figure 10, Evaluation Metrics for Traditional System

Figure 10 shows the evaluation metrics and the confusion matrix for the traditional gaze prediction system. The results of the confusion matrix show that the system produces a significant number of misclassifications and almost fails to correctly predict the center gaze direction. This system also fails to predict any gaze direction on 117 images out of 959 that

are used for testing. However, the system performs well in identifying the right gaze direction.

The evaluation metrics show that only 53% of the predictions were correct. A precision value of 66% reflects that the system is reliable in predicting positive predictions but a recall of 53% shows that the system struggles to identify the actual positive cases. This imbalance between the precision and recall value results in a lower F1-Score of 49%.

6.2 Experiments with Direct Approach CNN Model

The direct CNN approach built for this system involved directly feeding the face images into the model to produce gaze predictions. For the training, the number of epochs used were 10 and a batch size of 16 was used. However, this approach struggled to learn the complex facial patterns from the data and failed to learn the gaze features.

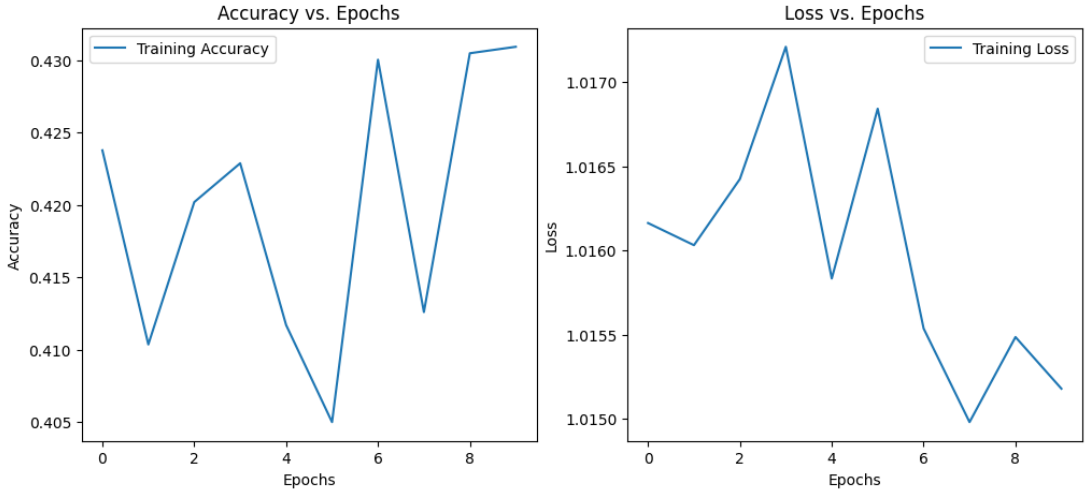

Figure 11, CNN Direct Approach Evaluation Graphs

Various fine-tuning steps, including adjustment of filters in layers, use of different optimiser functions, and adjustment of filter size, were conducted to expect better performance from the model. However, after a careful examination of the accuracy and loss plots shown in Figure 11, it was realised that the model completely fails to understand the expected outcome from it. Further analysis of the issue revealed that eye regions in face images occupy a very small area relative to the image, making it complex for the model to learn the critical features for gaze detection. For this reason, a pipelined model was developed to address the issues faced by the direct approach model.

6.3 Experiments with Pipelined CNN Model

Since this model works with a junction of two CNN models, each model was trained separately on the same test sets to achieve consistent evaluation. Several experiments were conducted with different numbers of epochs, kernel sizes, and dropout rates. For this model, a batch size of 16 was used to achieve a balance between model efficiency and training time. The final models for each of the modules were then used in the pipelined model for a detailed evaluation. The below sections explain the metrics and visualisations for the modular and the final pipelined model.

Eye Localization Model

Filter Size	Epochs	IoU (training)	Loss (training)
3x3	10	0.43	88.7
3x3	13	0.51	85
7x7	5	0.12	2200
7x7	15	0.55	67.2

Gaze Prediciton Module

Filter Size	Epochs	Accuracy (training)	Loss (training)
8x8	10	91%	0.23
8x8	5	80%	0.8
3x3	10	89%	0.26
3x3	5	80%	0.78

Figure 12, Experiments with Pipelined CNN Model

The set of parameters used in the experiments for the modular models is shown in Figure 12, which represents the metrics achieved by the modular models with the parameters that achieved the best result highlighted in the image. In these experiments, the filter size and number of epochs were tuned, and the rest of the parameters, such as the optimisation function and number of layers, were kept the same as these were implemented in the model architectures recommended by the previous research.

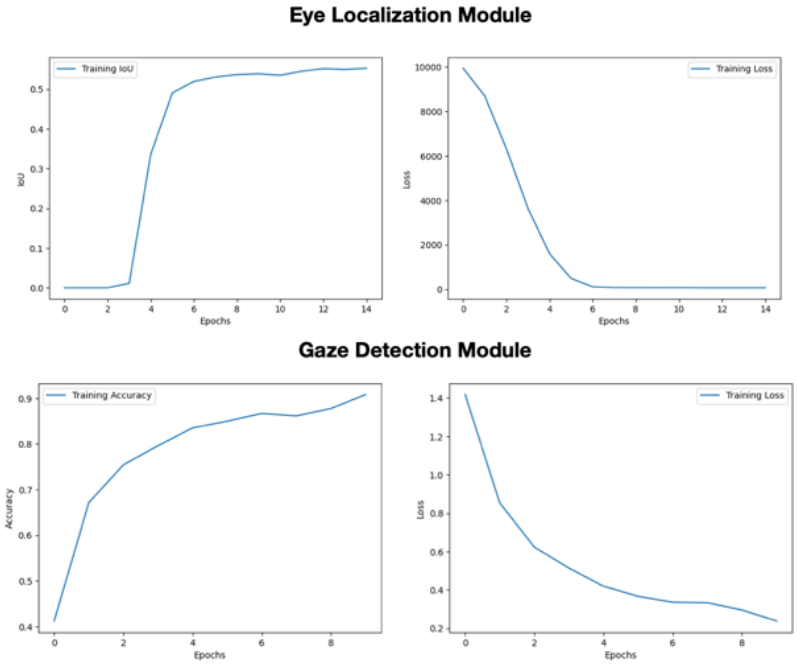

Figure 13, Training Performance for Pipelined CNN Modules

The visualisations in Figure 13 show the performance of the eye localisation and gaze detection modules during the training. The eye localisation module shows a significant increase in IoU during the initial epochs, with the training loss decreasing sharply, showing that the model quickly learns to localise the eye regions. However, the gaze detection module shows a gradual and steady increase in accuracy, and a smooth decrease of loss, which indicates that the model is optimised consistently without any regularities. These visualisations reflect that a stable training process was performed, unlike the direct approach model, which failed to achieve this. These results further validate the decision to train the models separately, enabling each module of the pipeline to focus on specific tasks.

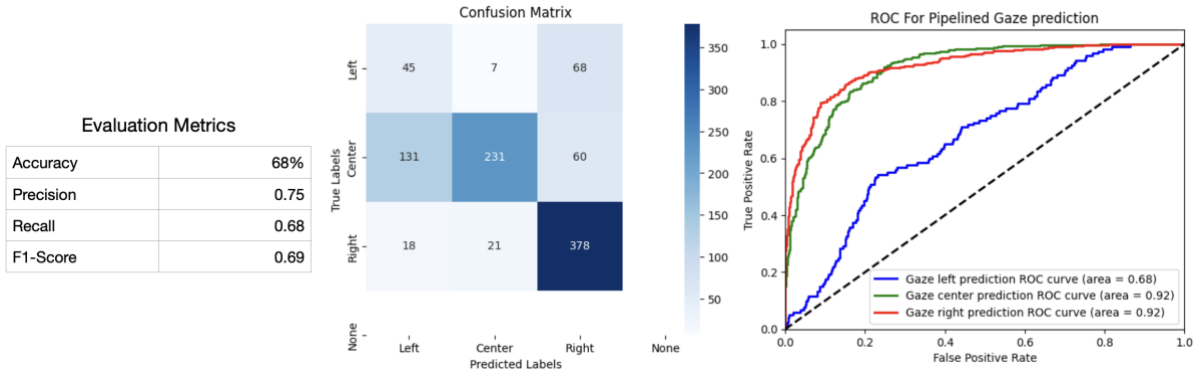

Figure 14, Testing Performance of Pipelined CNN model

The metrics and visualisations in Figure 14 represent the testing performance of the pipelined model. Starting with the evaluation metrics, an accuracy value of 68% reflects the success of the model in predicting gaze directions. A value of 0.75 for the precision metric shows that the predictions made by the model are accurate with low false positives. Further, a recall value of 0.68 indicates a good proportion of true instances. However, the model struggles to predict the left gaze direction. Finally, the F1-Score value of 0.69 reflects a decent performance of the model, leaving some room for further improvement in generalisation.

The confusion matrix values and the ROC curve achieved by the model show that the model performs the best in classifying the right gaze direction and performs the weakest in predicting the left gaze direction. The ROC curve for the left gaze class further confirms that the model fails to identify the difference in the left gaze class. However, the model performs significantly well in predicting the centre gaze direction compared to the traditional gaze prediction, which failed to predict the centre gaze direction.

6.4 Discussion

The outcomes from the various experiments conducted in the previous section reflect many insights into the challenges and issues faced during the testing of traditional and advanced gaze prediction systems of this study. The results of the traditional system and the pipelined approach CNN model show that both models faced a significant challenge in predicting the left gaze direction. A critical inference of the results further shows that even though the pipelined approach did not fully succeed in learning the critical features for gaze prediction, utilising a pipelined approach suggested by (Herashchenko & Farkaš, 2023) was a beneficial decision. This is confirmed by the issues faced during the implementation of the direct CNN model, which highlights a potential issue with convolutional neural networks in focusing on small regions of interest, i.e. eye regions in face images. Similarly, the results of the traditional gaze prediction system show the inability of facial computer vision algorithms to identify face and eye regions due to head poses.

To improve the research and design of this study, future work could focus on implementing robust augmentation techniques to overcome the issue of misclassifications and an end-to-end pipelined model to minimise the error propagating in the pipeline, as shown by a low accuracy during the final test of the pipelined model. Further, the use of large models such as vision transformers incorporating attention mechanisms can be utilised to

emphasise eye regions in face images to improve the model's ability to understand gaze patterns.

In the context of the previous research, it was seen that the modular approach aligns with the existing literature, which emphasises separating facial features for accurate prediction of gaze directions. Similarly, the issue of misclassifications reflects the previous research, where pipelined models struggle to differentiate among different gaze directions due to the small size of the dataset. The findings of this study contribute significantly to the field of gaze prediction by highlighting the strengths and limitations of different approaches used to develop gaze prediction, therefore providing valuable insights to future researchers.

7 Conclusion and Future Work

The motive of this research was to identify the strengths and limitations of advanced gaze prediction systems and traditional gaze prediction systems to provide recommendations to future researchers and developers for selecting a particular technique in their gaze prediction applications. To achieve this, various experiments were conducted using different techniques developed for this study.

The findings of this study highlight that CNN can be used in environments where frequent head poses are present in facial images, such as during driver monitoring, in conditions with various lighting, and in environments where there is a critical need for high accuracy in predictions such as medical diagnosis, therapy, and surveillance. On the other hand, traditional systems can be used where the system needs to work on small-scale hardware, such as VR glasses, where individuals do not show frequent or no head-poses, such as pointer control in VR headsets and online exam monitoring where candidates are looking straight towards the screen.

This study further reveals several insights for each gaze prediction system. It demonstrates the efficiency of the pipelined approach in improving feature extraction for complex tasks, which can encourage the research and development of pipelined gaze prediction systems. However, the limitations of this study, such as the preprocessing stage and imbalance in the predicted gaze direction reveal a need for improvement. Future work can explore the usage of vision transformers, multiple datasets, and cross-validation of dataset labels using different methods to minimise errors produced during model development.

There are many commercial applications, such as virtual and augmented reality, medical diagnosis software, and virtual reality, which utilise gaze prediction as a core feature. The systems developed for this study can be deployed to these applications with modifications and fine-tuning to meet the specific requirements of each application. Future research can work on refining these systems to achieve better performance in specific applications, reliable real-time performance, and increased generalizability to serve a wider range of execution environments.

References

Al Hamdi, M. J., Mumtaz, W., Albar, N., Mifta, T. A., Ridha, J., & Muchtar, K. (2023). A Low-cost Raspberry Pi and Deep Learning System for Customer Analysis. 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), 328–329.

Ansari, M. F., Kasprowski, P., & Obetkal, M. (2021). Gaze Tracking Using an Unmodified Web Camera and Convolutional Neural Network. Applied Sciences, 11(19), 9068.

Cao, X. (2023). Eye Tracking in Human-computer Interaction Recognition. 2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE), 203–207.

Chandel, S., Bhattacharya, R., Nayak, M., & Pathak, A. (2024). Integrating Eye Gaze Estimation with the Internet of Medical Things (IoMT) for Individualized and Efficient Healthcare. 2024 2nd World Conference on Communication & Computing (WCONF), 1–6.

Ewaisha, M., Shawarby, M. E., Abbas, H., & Sobh, I. (2020). End-to-End Multitask Learning for Driver Gaze and Head Pose Estimation. Electronic Imaging, 32(16), 110-1-110–116.

Herashchenko, D., & Farkaš, I. (2023). Appearance-Based Gaze Estimation Enhanced with Synthetic Images Using Deep Neural Networks. 2023 IEEE Symposium Series on Computational Intelligence (SSCI), 129–134.

Hu, Z., Li, S., Zhang, C., Yi, K., Wang, G., & Manocha, D. (2020). DGaze: CNN-Based Gaze Prediction in Dynamic Scenes. IEEE Transactions on Visualization and Computer Graphics, 26(5), 1902–1911.

K R, S., R, P., N.S, Aishwariya., & Rathishree, M. (2023). Cursor Control Based on Eyeball Movement Using Deep Learning. 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), 1–5.

Kar, A., & Corcoran, P. (2017). A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms. IEEE Access, 5, 16495–16519.

Kar, A., & Corcoran, P. (2018). Gaze Visual—A Graphical Software Tool for Performance Evaluation of Eye Gaze Estimation Systems. 2018 IEEE Games, Entertainment, Media Conference (GEM), 1–9.

Kathpal, K., Negi, S., & Sharma, S. (2021). iChat: Interactive Eyes for Specially Challenged People Using OpenCV Python. 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 1–5.

Lemley, J., Kar, A., Drimbarean, A., & Corcoran, P. (2019). Convolutional Neural Network Implementation for Eye-Gaze Estimation on Low-Quality Consumer Imaging Systems. IEEE Transactions on Consumer Electronics, 65(2), 179–187.

Li, J., Ray, S., Rajanna, V., & Hammond, T. (2022). Evaluating the Performance of Machine Learning Algorithms in Gaze Gesture Recognition Systems. IEEE Access, 10, 1020–1035.

Li, Y., & Li, X. (2024). EDUNet: Enhancing Detail Awareness in U-Shaped Network for 2D Human Pose Estimation. 2024 3rd International Conference on Artificial Intelligence and Computer Information Technology (AICIT), 1–4.

Maralappanavar, S., Behera, R., & Mudenagudi, U. (2016). Driver's distraction detection based on gaze estimation. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2489–2494.

Miah, P., Gulshan, M. R., & Jahan, N. (2022). Mouse Cursor Movement and Control using Eye Gaze- A Human Computer Interaction. 2022 International Conference on Artificial Intelligence of Things (ICAIoT), 1–6.

Mounica, M. S., Manvita, M., Jyotsna, C., & Amudha, J. (2019). Low Cost Eye Gaze Tracker Using Web Camera. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 79–85.

Prince, M., Santhosh, N., Thankachan, N., Sudarsan, R., & Anjusree, V. K. (2020). Eye Movement Classification Using CNN. 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), 138–142.

Ryu, H. S., Ju, U., & Wallraven, C. (2024). Predicting Future Driving Decisions in an Accident Situation From Videos: A Combined Behavioral, Eye Gaze, and Computational Analysis. 2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), 391–392.

Samaniego, M., Atxa, E., Rodriguez, I., & Lazkano, E. (2024). Pepper says: "I spy with my little eye." 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids), 53–59.

Smith, B. A., Yin, Q., Feiner, S. K., & Nayar, S. K. (2013). Gaze locking: Passive eye contact detection for human-object interaction. Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, 271–280.

Viswanatha, V., Ramachandra, A. C., Reddy, G. L., Reddy, A. V. S. T., Reddy, B. P. K., & Kiran, G. B. (2023). An Intelligent Camera Based Eye Controlled Wheelchair System: Haar Cascade and Gaze Estimation Algorithms. 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), 1–5.

Xiang, Y., Yang, H., Hu, R., & Hsu, C.-Y. (2021). Comparison of the Deep Learning Methods Applied on Human Eye Detection. 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 314–318.

Yuli Cristanti, R., Sigit, R., Harsono, T., Adelina, D. C., Nabilah, A., & Anggraeni, N. P. (2017). Eye gaze tracking to operate android-based communication helper

application. 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 89–94.

Zhu, A., Yang, J., & Yu, W. (2021). A novel target tracking method of unmanned drones by gaze prediction combined with YOLO algorithm. 2021 IEEE International Conference on Unmanned Systems (ICUS), 792–797.