CUSTOMER RETENTION FOR E-COMMERCE USING GAZE TRACKING

NAME: DHRUV ROHATGI

DEPARTMENT: SCHOOL OF COMPUTING

ROLL NO: 210441356

SUPERVISOR: PROF. BOGUSLAW OBARA

WHAT IS GAZE TRACKING?

- Gaze Tracking, also known as eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head.

Eye Tracker

- An eye tracker is a device for measuring eye positions and eye movement. The most commonly used setup for eye tracking is a web camera connected to a device.

HOW IS GAZE TRACKING DONE?

- Skin electrodes placed around the eyes
- Head-mounted eye trackers
- 2D regression model-based method
- 3D eye model-based method
- Appearance-based methods

2D MODEL BASED METHODS

About

- 2D model-based method directly maps the feature vector to the point of gaze (POG) using the transformation function.

Drawback

- Dedicated complex setup such as infrared cameras are required.

3D MODEL BASED METHODS

About

- The 3D model-based method constructs a geometric model of the eyes to estimate gaze

Drawback

- Dedicated complex setup such as infrared cameras are required

CNN BASED METHODS

About

- 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.

APPEARANCE-BASED GAZE ESTIMATION METHODS

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.

ADVANTAGES OF CNN BASED SOLUTION

It can extract high dimensional eye features from high dimensional image data.

It can learn a highly nonlinear function to estimates the gaze.

TYPES OF EYE MOVEMENT

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 behaviour
Gaze Duration	Cognitive processing, conveying intent	Item selection, text/number entry
Blink	Indicates behavioural states, stress	Eye liveliness detection, activate command
Pupil Size Change	Cognitive effort, representing micro emotions	Assess cognitive workload, user fatigue, command

SETUP AND METHOD FOR ESTIMATING EYE GAZE

- Require: Digital Camera/ Near Infra-Red (NIR) LED's, Computer
- Steps: User Calibration Screen, Obtaining Video Frames(Face/Eyes), Detecting Eyes, Mapping Gaze Coordinates on Screen, Gaze Computed from Relative Movement between pupil Center and Glint Position.

USER INTERFACE FOR GAZE TRACKING

- Active: Users gaze information is used as an input modality and to activate a function
- Passive: ye gaze data is consolidated to predict user interest or attention
- Single: Only user's gaze is used as an input variable
- Multimodal: Gaze input is combined with a mouse, keyboard, touch, or blink inputs for command in a multimodal interface

ESTIMATION OF GAZE TRACKING ACCURACY

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

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

Gaze Point Coordinates in Pixels

In the literature, gaze tracking accuracy measures are reported in different ways,

- angular accuracy in degrees
- distance accuracy in cm/mm
- distance in pixels
- gaze estimation accuracy in percentage

DATA PREPARATION

DATA COLLECTION

DATA CALIBRATION

DATA PREPROCESSING

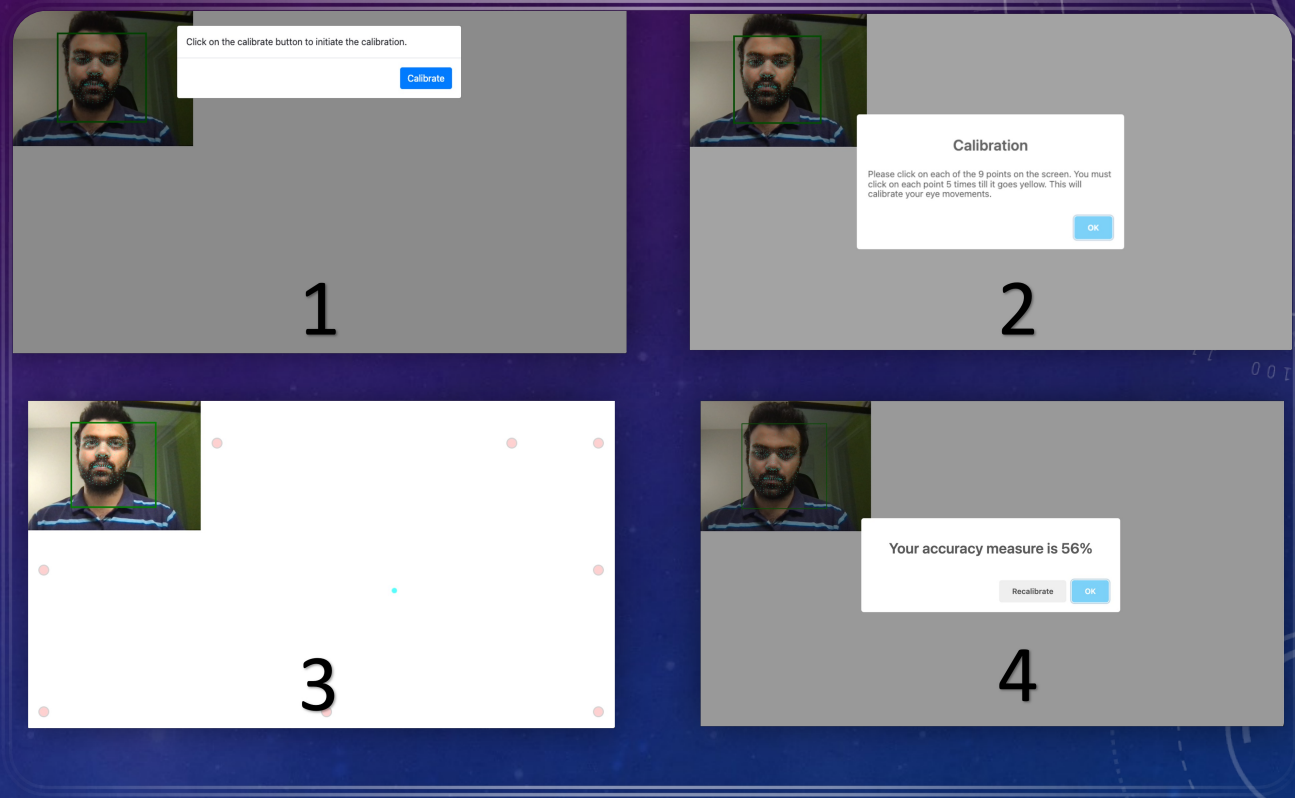
LIGHTING NORMALISATION

DATA COLLECTION METHODS

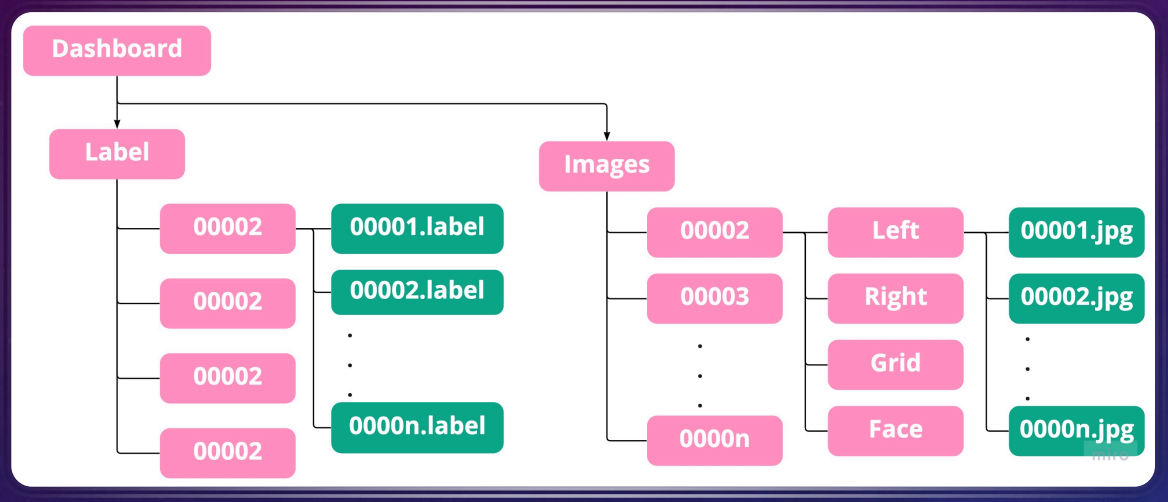
- To collect the required data for gaze tracking, a calibration window is created.
- The objective was to collect the images of the users while using the application as they would be doing activities of daily life while using a laptop.
- Users were shown the red dots on the screen as shown in the figure.

DATA CALIBRATION

- 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.



DATA PREPARATION



- All the collected images were resized to 512 X 512 pixel.
- The resized images are passed 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 recognized 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.

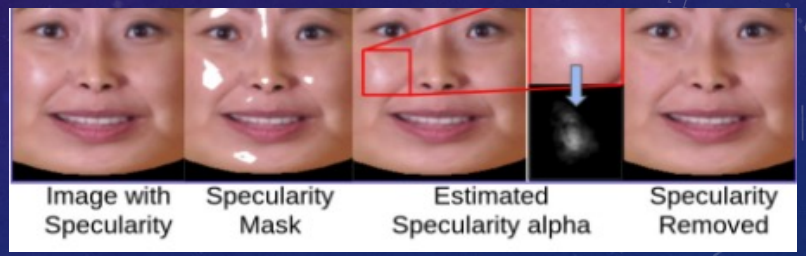
LIGHTING NORMALISATION ON THE DATA

An image is taken as input to remove the specaluratiy.

The input image is converted to grey scale.

$$I = \alpha + (1 - \alpha) * I_c$$

The specular image is modeled using the above equation



Smoothing the input image and eliminating pixels with RGB values above 90 removed specularity. Images were filled with neighbouring pixel values using whole filing.

IMPLEMENTATION & DISCUSSION

Experiment

Training

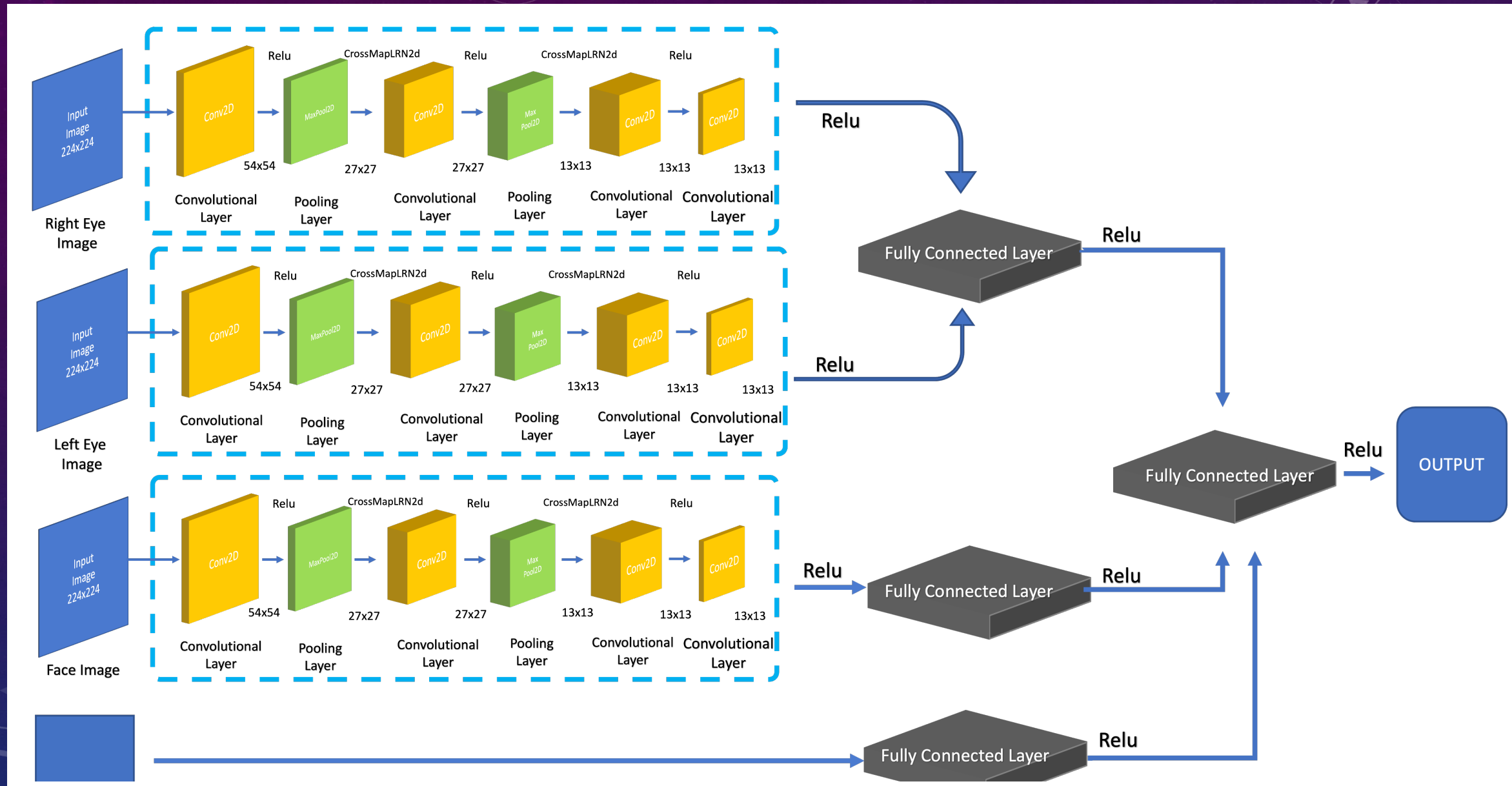
Evaluation

EXPERIMENT

Proposed a solution that uses appearance-based gaze estimation, convolutional neural network architecture. The model used in this experiment comprises three overall networks concatenated at the end.

Following are the three networks used for creating the model:

1. Left and Right Eye
2. Face
3. Grid


Input Image 224x224
Right Eye Image
Conv2D
Relu
MaxPool2D
CrossMapLRN2d
Max Pool2D
54x54
27x27
13x13
Convolutional Layer
Pooling Layer
Left Eye Image
Face Image
Fully Connected Layer
OUTPUT

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.

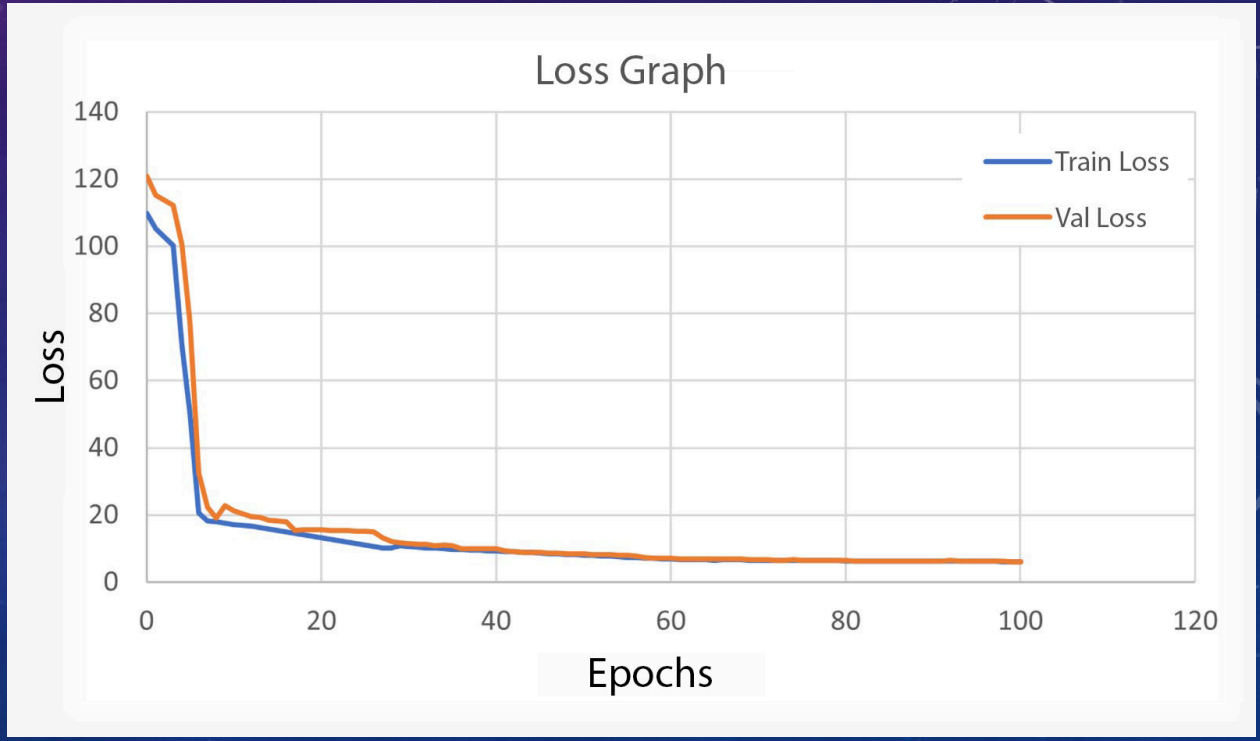
$$MSE = 1/n \sum_{n} (y_i - \tilde{y}_i)^2$$

Mean Square Error

EVALUATION

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

The best loss is 6.23080015



CUSTOMER RETENTION- APPLICATION

The project suggests a Gaze Tracking Implementation for Customer Retention on eCommerce.

On collecting insights from gaze, one may be able to get useful insights from this data.

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 point and time.
3. Next time, make better search results and advertisements that the user could find interesting using the data acquired.

CONCLUSION AND FUTURE WORK

- 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.

Future Work

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.