International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 13 Issue 02 February 2026

Evaluating the Performance of an Eye-Gaze Tracking System Using the VGG16 Deep Learning Technique

Manisha Chouhan[1], Mohit Jain[2]
Research Scholar[1], HOD CSE[2]
BM College of Technology, Indore (M.P.), India[1, 2]
chmanisha72@gmail.com[1], bmctmohitcs@gmail.com[2]

Abstract: *Eye gaze tracking systems have become increasingly important due to their diverse applications across various fields. This study introduces an innovative eye gaze tracking system utilizing the VGG16 algorithm, designed to precisely estimate and track the direction of an individual's gaze based on their eye movements. The system follows a multi-step process, including pre-processing, region of interest (ROI) segmentation, feature extraction, and training with VGG16 for gaze tracking. The proposed system addresses challenges such as variations in eye appearance, head movements, real-time performance, and adaptability, through effective calibration and personalization. This system has significant potential in areas like assistive technology, human-computer interaction, healthcare, psychology, and more. To evaluate the system's effectiveness, parameters like accuracy, precision, real-time performance, and usability will be measured. The findings from this study are expected to offer valuable contributions to the advancement of eye gaze tracking technology and its broad range of applications.*

Keywords: *Eye Gaze tracking, Convolutional Neural Network (CNN), VGG16,deep learning*.

1. INTRODUCTION

The major innovation in eye tracking was the invention of a head-mounted eye tracker [1] this technique is still widely used (Figure 1). Another reference work in the gaze tracking world is the one done by Yarbus. Yarbus was a Russian psychologist who studied eye movements and saccadic exploration of complex images in the 1950s and 1960s. He recorded the eye movements performed by observers while viewing natural objects and scenes [2]. Here again, this work tends to show that the gaze direction is crucial in interactivity, actually [3] showed that the gaze trajectories followed depends on the task that the observer has to perform. This section presents in a concise way different application that use eye and gaze tracking systems, and how they take advantage of the data collected. Historically, the first application using eye tracking systems was, as we already saw, the user interface design. Actually, knowing where people are looking at is really important to design devices, cockpits, cars.

Figure 1: A non intrusive eye tracking system.

Many Publications can be found relating to this field from [4] to [5] .All these studies are based on the eye-mind hypothesis that what a person is looking at is assumed to indicate the thought on top of the stack of cognitive processes. In this way, the visibility, meaningfulness and placement of specific interface elements can be objectively evaluated and the resulting findings can be used to improve

the design of the interface. A huge range of works that inherits directly from the previous is the human-computer interaction (HCI), but currently very few applications have been implemented for consumer products. The main reason for the slow emergence of attentive interfaces utilizing eye gaze is what dubbed the Midas touch problem: the application should not react every time the target of the gaze changes, only in appropriate situations, and at the right moment, very complex notions for a computer.

Even if commercial applications are quite uncommon, a key application for eye tracking systems is to enable people with severe physical disabilities to communicate and/or interact with computer devices. Simply by looking at control keys displayed on a computer monitor screen, the user can perform a broad variety of functions including speech synthesis, control, playing games, typing. A network of excellence, supported by the European Commission's IST 6th framework program, was created around this topic: the COGAIN network (Communication by Gaze Interaction www.cogain.org). On this website, interested readers could found many related publications. Eye tracking systems can enhance the quality of life of a disabled person, his family and his community by broadening his communication, entertainment, learning and productive capacities. Additionally, eye tracking systems have been demonstrated to be invaluable diagnostic tools in the administration of intelligence and psychological tests [6].

Another aspect of eye tracking usefulness could be found in the cognitive and behavioral therapy, a branch of psychotherapy specialized in the treatment of anxiety disorders like phobias. Technology offers a tool for quantitatively measuring and recording what a person does with his eyes as he reads. At the opposite of helping people with disabilities, other applications include military weapon control and remote robotics. As we saw, militaries sponsored much of the early research and development on eye tracking systems, and one of their primary objectives was to aid pilots, busy flying the plane, in their weapons control. Eye tracking systems allow the pilots to observe and select targets with their eyes while flying the plane and firing the weapons with their hands. Another aspect of eye control is reached through vehicle simulation. Among all the eye tracking studies those concerning vehicle simulation is probably the most numerous, attention, tiredness, dangerous behaviour. for a complete view of simulation aspect, readers may found information via the ITS Institute's primary human factors research laboratory web site (http://www.humanfirst.umn.edu/).

Finally, we want to emphasize the current and prospective aspect of eye and gaze tracking in game environment, either in rehabilitation, an entertainment or an edutainment context; actually, it is mentioned in a very interesting report done in 2005 [34] that : .no information about eye tracking and computer games was found. Nevertheless, some previous work was interesting to mention relating to eye behaviour in a game environment, to make possible to lay out the foundations of player's eye behaviors [7]. The research suggests that action game playing might be a useful tool to rehabilitate visually impaired people or improve visual attention. Recently, a Queens University study confirms that video-gamers feel more immersed and have more fun in virtual environments when they play with commercial eye tracking technology. Moreover, the growing of video game enthusiasts to leave the real world in favor of a virtual one is driving a market trend toward developing [8].

2. LITERATURE REVIEW

Prakash Kanade, et.al.[9] To avoid the rising number of car crash deaths, which are mostly caused by drivers' inattentiveness, a paradigm shift is expected. The knowledge of a driver's look area may provide useful details about his or her point of attention. Cars with accurate and low-cost gaze classification systems can increase driver safety. When drivers shift their eyes without turning their heads to look at objects, the margin of error in gaze detection increases. For new consumer electronic applications such as driver tracking systems and novel user interfaces, accurate and effective eye gaze prediction is critical. Such systems must be able to run efficiently in difficult, unconstrained conditions while using reduced power and expense. A deep learning-based gaze estimation technique has been considered to solve this issue, with an emphasis on WSN based Convolutional Neural Networks (CNN) based system. The proposed study proposes the following architecture, which is focused on data science: The first is a novel neural network model that is programmed to manipulate any possible visual feature, such as the states of both eyes and head location, as well as many augmentations; the second is a data fusion approach that incorporates several gaze datasets.

Kara J. Emery et. al. [10] The widespread utility of eye tracking technology has created a growing demand for more consistent and reliable eye-tracking systems, and there is a need for new and accessible approaches that can enhance the accuracy of eye-tracking data. Previous studies have offered evidence for associations between certain non-eye signals and gaze such as a strong coordination between head motion and gaze shifts.

Zaryab Aslam et. al. 11] Face detection and eyes extraction has an important role in many applications such as

face recognition, facial expression analysis, security login etc. The system is designed to develop an eye-motion based communication system for patients suffering from motor neuron disease (MND) to contact with care takers whenever they want. In this paper, we proposed a method to efficiently track eye gaze detection in real time from video acquired by a web camera. The camera is stationary with respect to the head. Proposed system is mainly divided into three steps; face detection, eye-ball (pupil) detection and finally eye gaze detection. Using predefined facial landmarks face is detected in real time. Then in the second step face point that were extracted in phase one helps to detect eye point and using circular Hough Transform eye pupil was accurately detected. Finally, in the last stage eye gaze was calculated using gaze ratio formula. The experiments are conducted on people from different age groups and a high accuracy rate is achieved.

3. PROPOSED SYSTEM

The "EGY Gaze Tracking System" is a proposed system that aims to use the VGG16 model for gaze tracking. Gaze tracking involves estimating the direction of a person's gaze based on their eye movements or eye regions in images or videos. The VGG16 model is a deep learning convolutional neural network (CNN) that has been widely used for various computer vision tasks, including image classification and feature extraction. The proposed system likely follows these steps:

Data Collection: Collect a dataset of video containing eye regions with corresponding gaze direction labels (e.g., left, right, center).the dataset labeled and balanced to cover different gaze directions.

Data Preprocessing: Resize the eye images to a fixed size (e.g., 256x256) to match the input size of VGG16. Normalize the pixel values to a common scale (e.g., [0, 1]) for better convergence during training. Shuffle and split the dataset into training and testing sets.

Training VGG16: Initialize the VGG16 model with pre-trained weights on ImageNet or random weights. Fine-tune the VGG16 model on the training dataset using the following steps:

Forward Propagation: Feed the training eye images through the VGG16 layers to extract features.

Gaze Prediction Model: Add additional fully connected layers after the VGG16 layers to predict the gaze direction based on the extracted features.

Backpropagation: Compute the loss between predicted gaze direction and true labels and update the model's weights using gradient descent.

Epochs: Repeat the forward-backward process for multiple epochs to optimize the model

VGG16 Model: Use the pre-trained VGG16 model as the feature extractor. The model's weights have been learned on a large dataset to detect various visual features.

Feature Extraction: Extract features from the eye regions in the collected data using the VGG16 model. This can be done by taking the activations from a specific layer of VGG16.

Gaze Tracking:

- For each frame in the preprocessed video:
- Pass the frame through the trained VGG16 model:
- Extract features from the eye region using the VGG16 layers.
- Feed the extracted features into the gaze prediction model to obtain the predicted gaze direction.
- Visualize the gaze direction on the frame (e.g., drawing an arrow pointing towards the predicted gaze direction).

Training: Train the gaze prediction model using the extracted features and corresponding gaze direction labels from the collected dataset.

Evaluation: Evaluate the performance of the gaze tracking system on a separate test dataset. Compute metrics such as accuracy; mean squared error, other relevant metrics to assess the system's performance.

VGG16 algorithm

The VGG16 algorithm is a deep convolutional neural network (CNN) that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is characterized by its depth and simplicity, consisting of 16 layers (hence the name "VGG16"). In this step-by-step description, I'll explain the key components and mathematical operations involved in the **VGG16 algorithm.**

Step 1: Input Image

- The input to VGG16 is a colored image (usually with three color channels: Red, Green, and Blue).

Step 2: Convolutional Layers

- VGG16 consists of multiple convolutional layers. Each convolutional layer is responsible for detecting different patterns or features in the input image.
- A convolutional layer involves the following mathematical operations

Convolution: A set of learnable filters (kernels) is convolved with the input image to produce feature maps. Each filter detects specific patterns (edges, textures, etc.) in the image.

Activation: After convolution, an activation function (commonly ReLU, Rectified Linear Unit) is applied element-wise to introduce non-linearity.

Pooling: A max-pooling operation is performed to downsample the feature maps, reducing the spatial dimensions and the number of parameters in the network.

Step 3: Fully Connected Layers

- After several convolutional and pooling layers, the output is flattened into a 1D vector. This vector is then fed into fully connected layers to make the final predictions.
- A fully connected layer involves the following mathematical operation:
- Matrix Multiplication: The flattened feature vector is multiplied by a set of learnable weights and biases to produce the final output, which represents class scores or probabilities.

Step 4: Softmax Activation

The final layer of VGG16 applies the softmax activation function to convert the class scores into probabilities. Softmax ensures that the output probabilities sum up to 1, making it suitable for multiclass classification problems.

Step 5: Training

During the training phase, VGG16 is fed with labeled images. It learns the optimal filter weights and biases through a process called backpropagation and gradient descent, minimizing a predefined loss function (e.g., cross-entropy loss).

Step 6: Inference

Once trained, VGG16 can be used for inference on new, unseen images. The network takes an input image, performs the forward pass (convolution, activation, pooling, fully connected layers, softmax), and outputs the predicted class probabilities.

VGG16 ARCHITECTURE

Input Layer: RGB Image

Convolutional Layers: Multiple layers with filters (kernels), ReLU activation, and max-pooling.

Fully Connected Layers: Flattening, matrix multiplication, and ReLU activation.

Output Layer: Softmax activation for classification.

Architecture Description

Input Image: $X \in \mathbb{R}^{(H \times W \times C)}$, where H is the height, W is the width, and C is the number of color channels.

Convolution Operation: $h_i = \sigma(\sum(X * W_i + b_i))$, where h_i is the output feature map, σ is the ReLU activation function, X is the input image, W_i is the i-th filter (kernel), b_i is the i-th bias, and * represents the convolution operation.

Pooling Operation: $h'_i = \text{MaxPooling}(h_i)$, where h'_i is the downsampled feature map obtained through max-pooling.

Fully Connected Layer: $Z = X * W + b$, where Z is the output vector, X is the input feature vector, W is the weight matrix, and b is the bias vector.

Softmax Activation: $P(\text{class}_i) = \exp(Z_i) / \Sigma(\exp(Z_j))$, where $P(\text{class}_i)$ is the probability of class_i, Z_i is the i-th element of Z, and Σ represents the sum over all elements in Z.

4. SIMULATION RESULTS

The code loads a video and reads its frames. Each frame is resized to a fixed size (256x256).

Face detection is performed using a cascade object detector, which draws rectangles around the detected faces. Eye detection is attempted on the detected face regions, drawing rectangles around the detected eyes. The processed video frames are displayed. It loads an image dataset from the 'TrainDataset' folder, assuming the folder structure represents different classes (labels). It counts the number of samples in each class to ensure balanced training data.

- It then splits the dataset into training and testing sets (70% training, 30% testing) using splitEachLabel function.
- The images are resized to a fixed size (256x256) and stored in augmentedImageDatastore for training and testing.

Training

- It loads the pre-trained VGG16 network using vgg16 function and analyzes its layers using analyze Network.
- The fully connected layer 'fc8' is used as the feature extraction layer.
- The features are extracted for the training set using activations function, and a linear support vector machine (SVM) classifier is trained using fitcecoc.

Testing

- The same feature extraction process is applied to the testing set using activations.
- The VGG16 classifier predicts the labels of the testing set.

Video Processing:

- The code allows selecting a video file and reads its frames.
- The frames are resized and stored in the 'Frames' folder.

- Face and eye detection is performed on each frame, and rectangles are drawn around the detected regions.

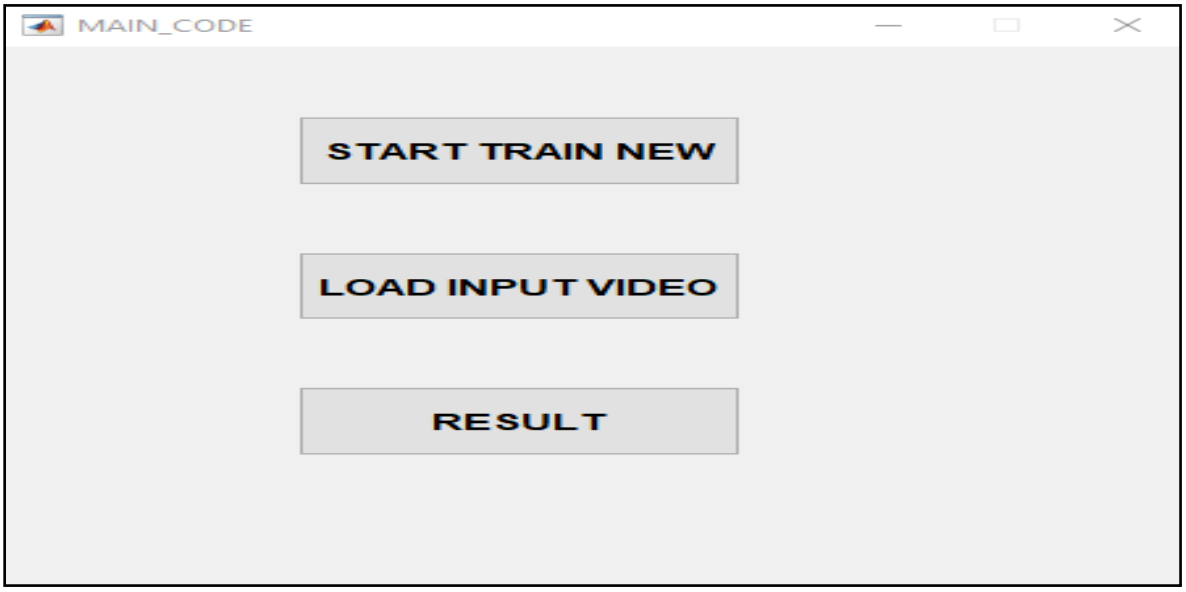

Figure 2: system flow diagram

The system flow diagram for the "EGY Gaze Tracking System" using VGG16, along with the training dataset, input video, and the expected results.

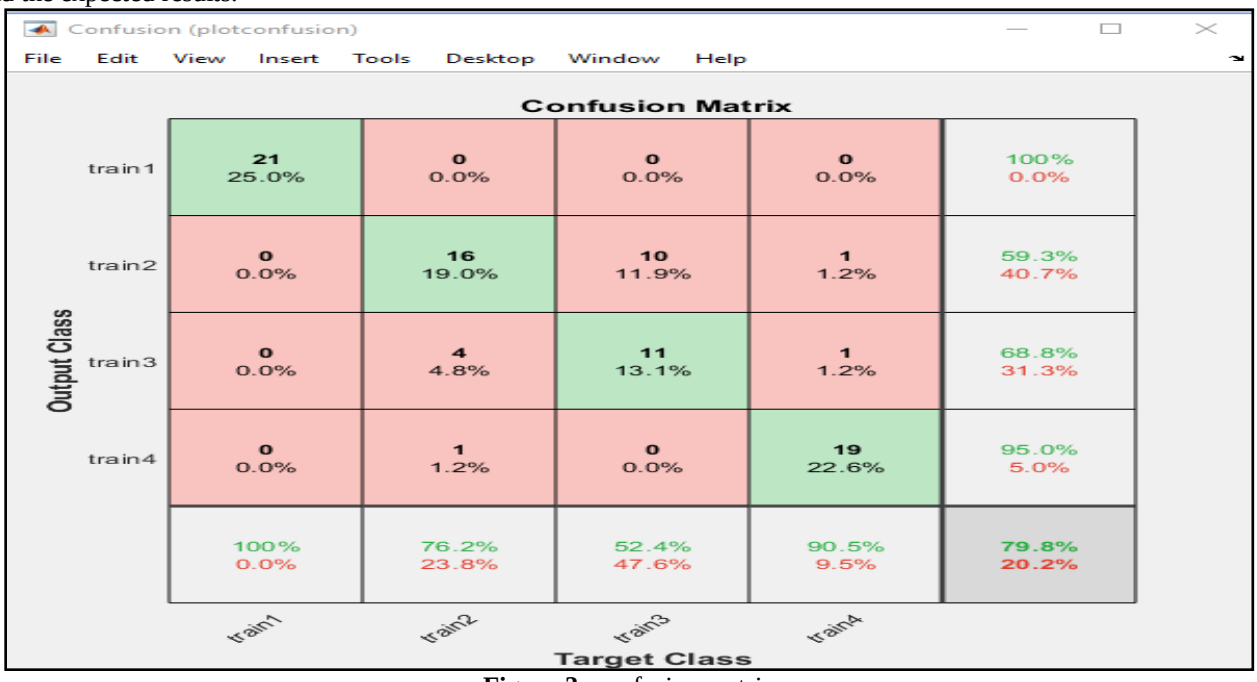

Figure 3: confusion matrix

A confusion matrix is a table used to evaluate the performance of a classification model. It shows the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model for each class.

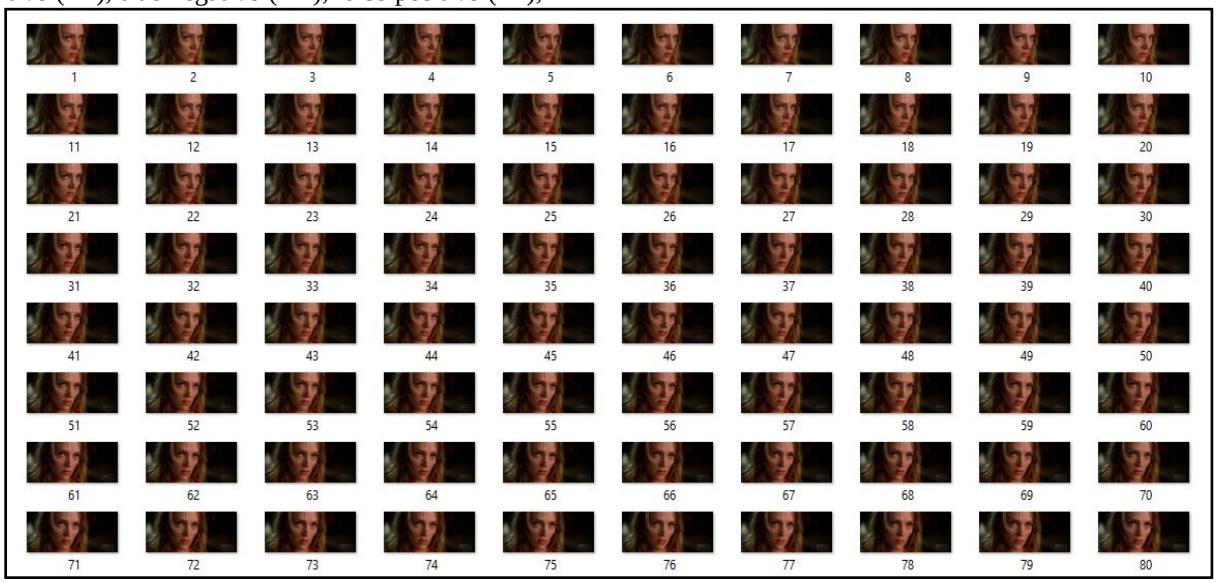

Figure 4: input video frame generation

Generating video frames from a video involves extracting individual frames from the video file.

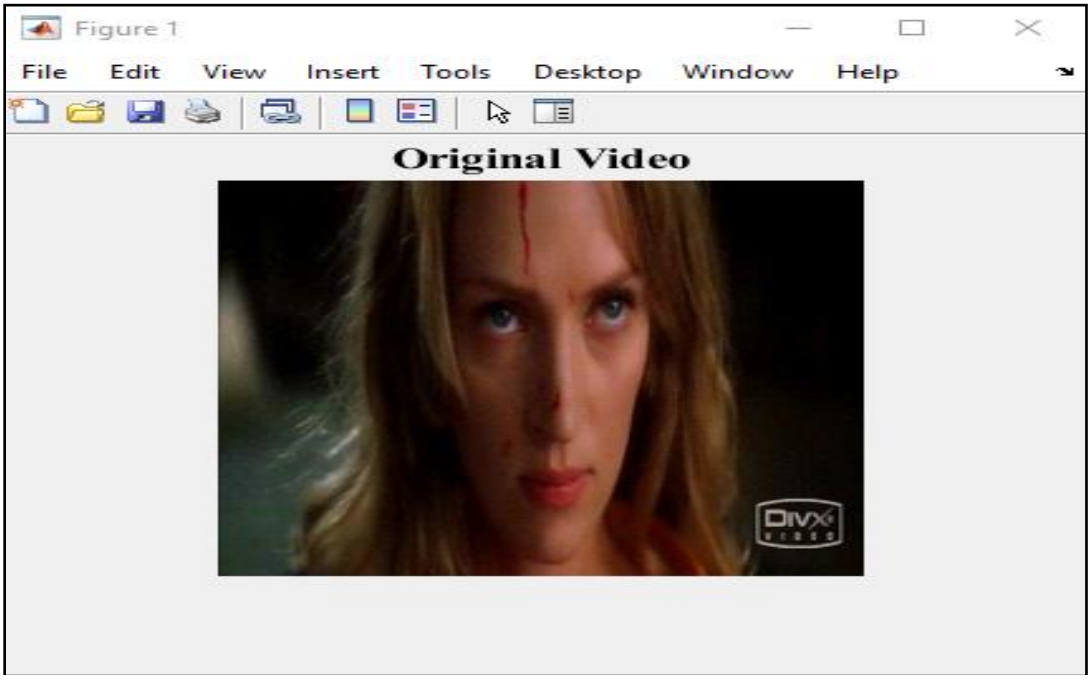

Figure 5: original input video

Preprocess each frame (resize, grayscale conversion, normalization, etc.) as needed for input to the VGG16 model.

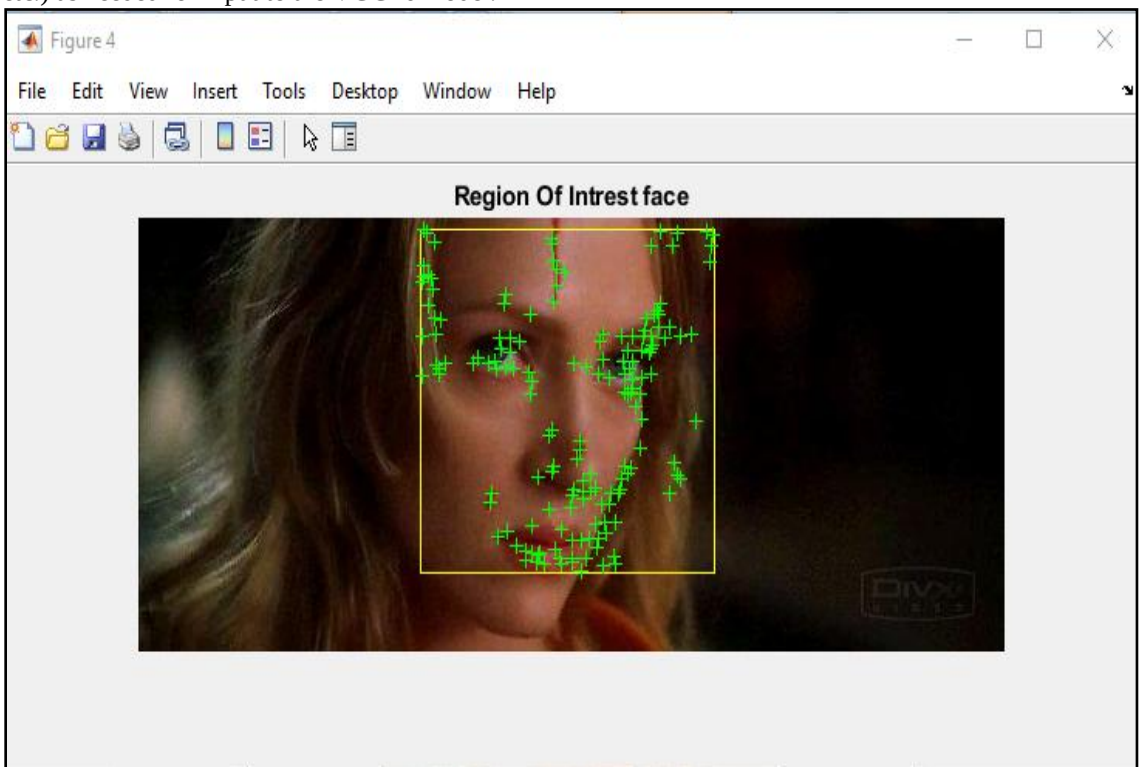

Figure.6 : region of interest

The area around the eyes, as the eyes contain important visual cues for determining the direction of a person's gaze. By selecting the region around the eyes as the ROI, the gaze tracking system can concentrate on extracting relevant features from that area and make more accurate gaze predictions.

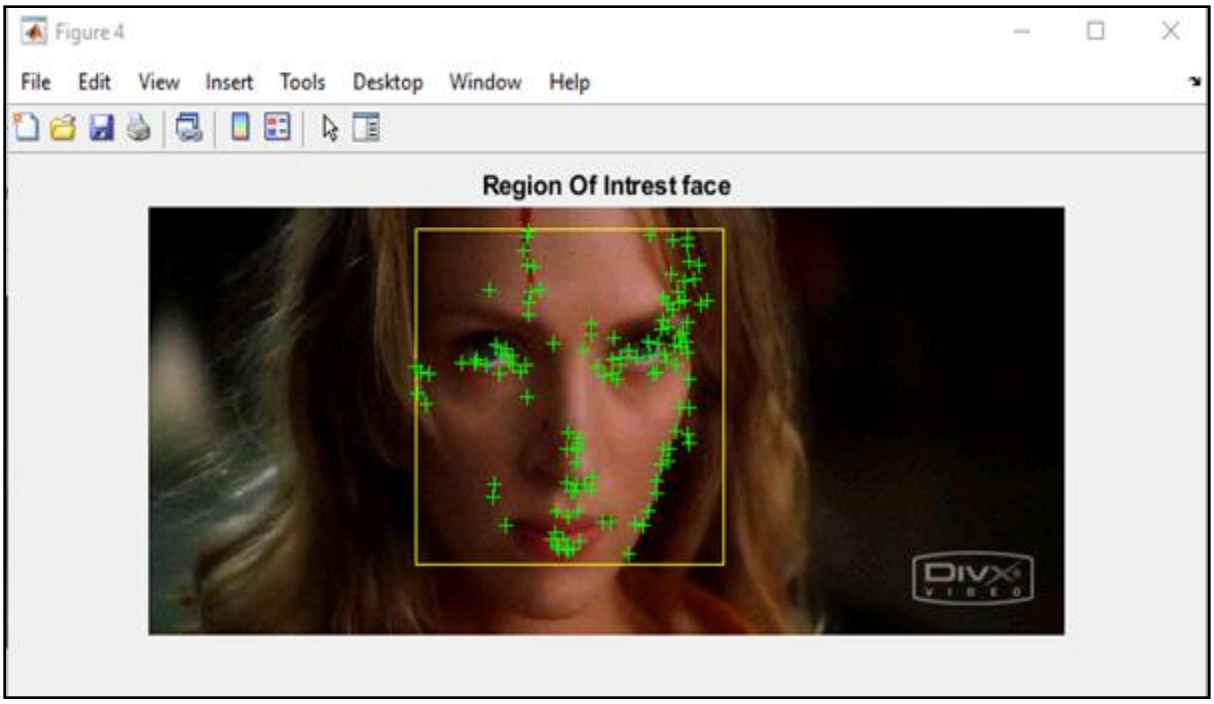

Figure 7: region of intersect with rotation frame

Use the tracked key points to calculate the new position and orientation of the ROI. Adjust the bounding box of the ROI based on the rotation and translation information obtained from optical flow.

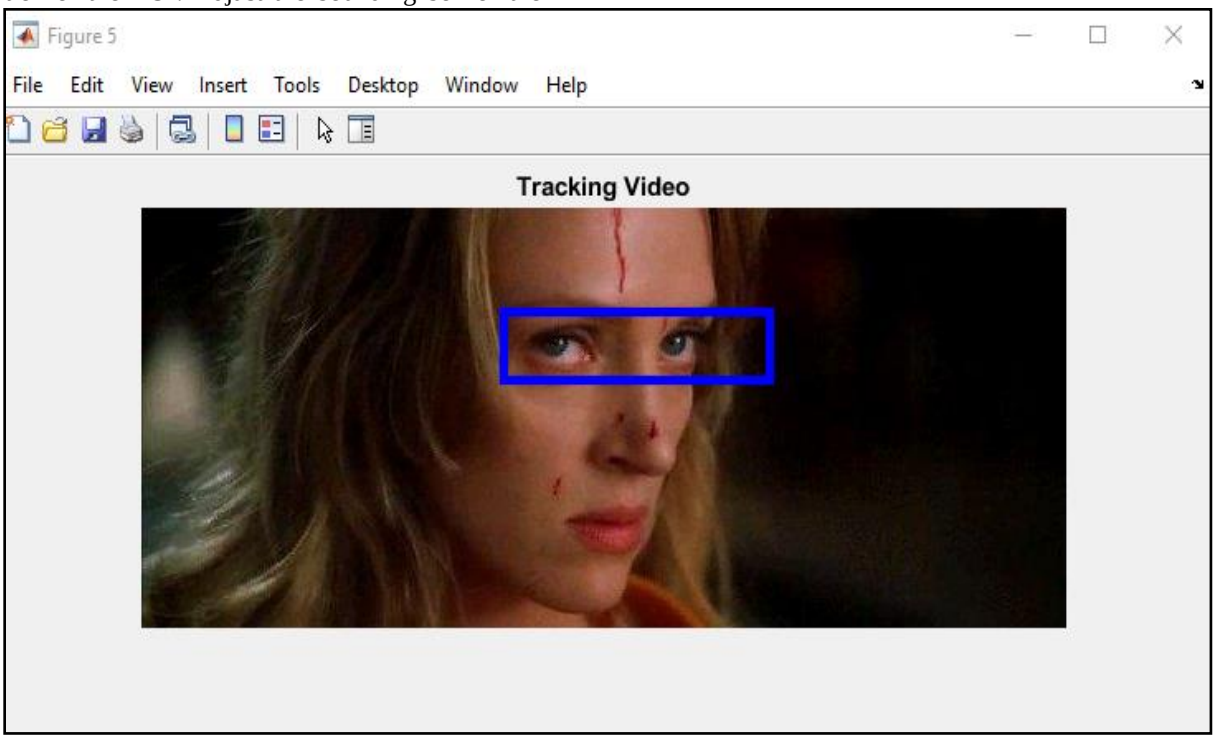

Figure 8: tracking video

Track objects or regions of interest (ROIs) across frames in a video.

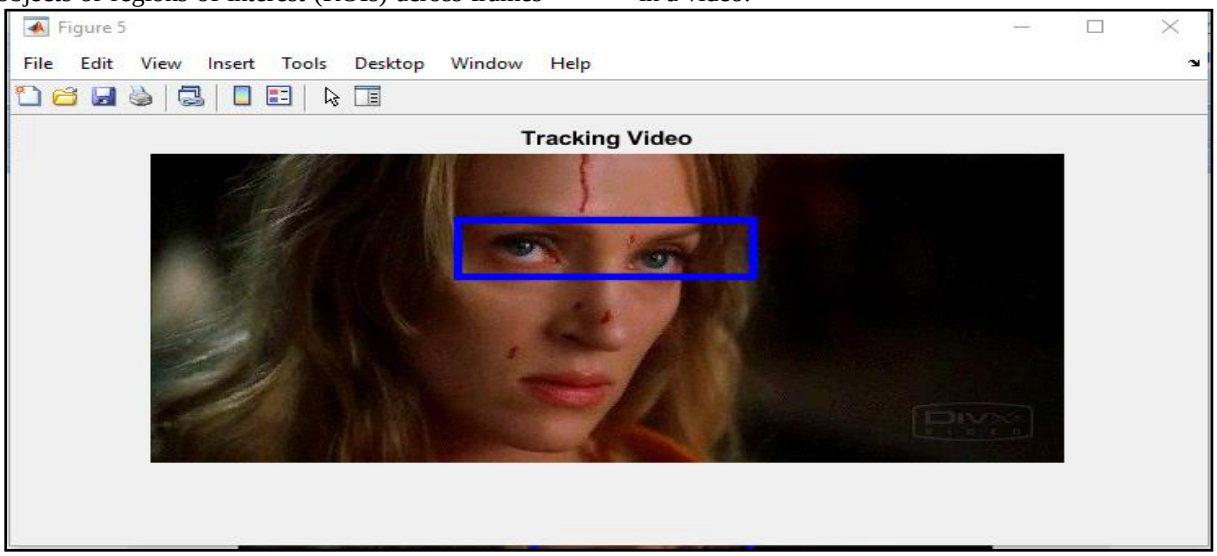

Figure 9: tracking video eye gaze

Feed the preprocessed eye ROIs through the gaze estimation model to obtain the gaze direction (e.g., left, right, up, down).

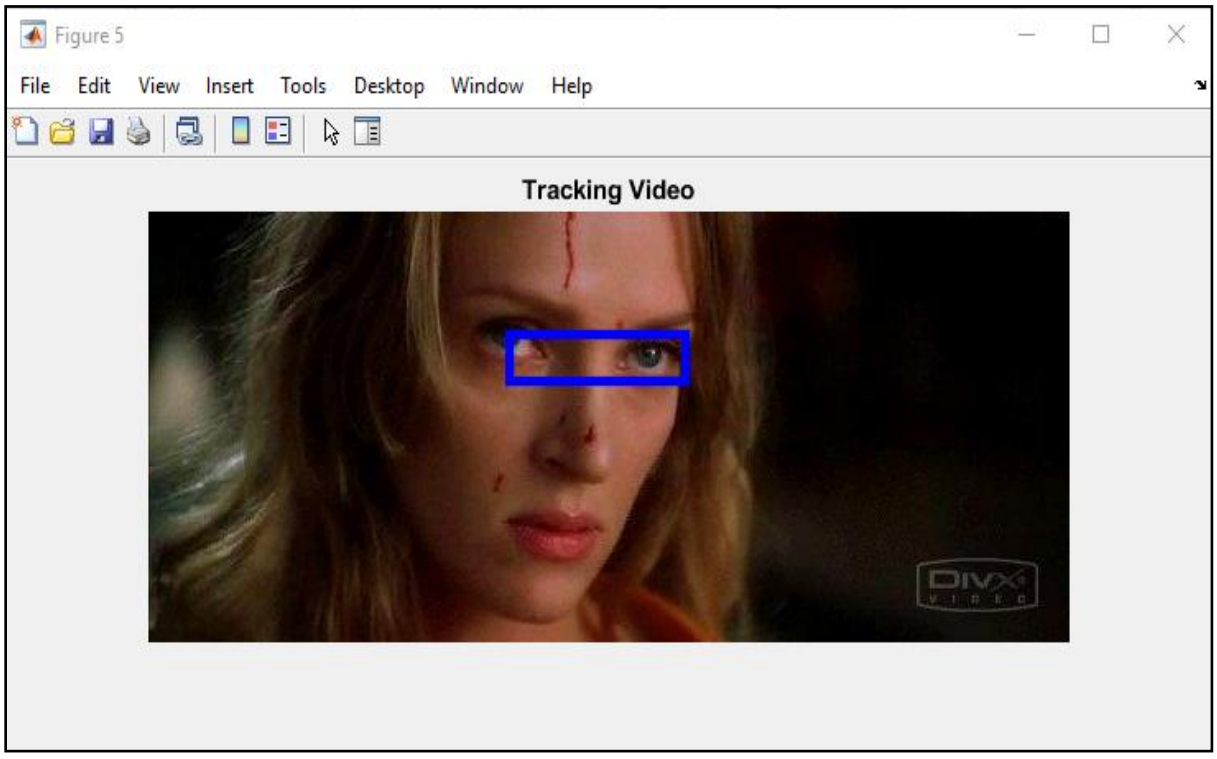

Figure 10: tracking video eye gaze detection

The process of face detection, eye detection, eye tracking, and gaze estimation for subsequent frames in the video to track the gaze over time.

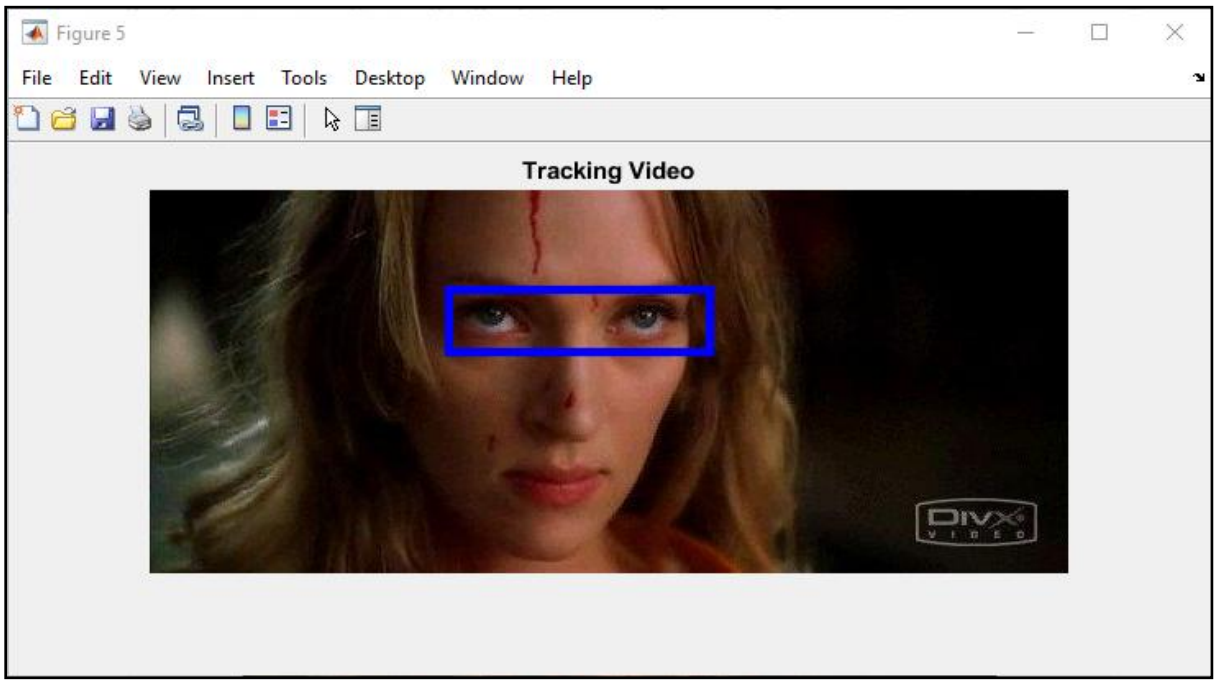

Figure 11: tracking video frame eye gaze detection

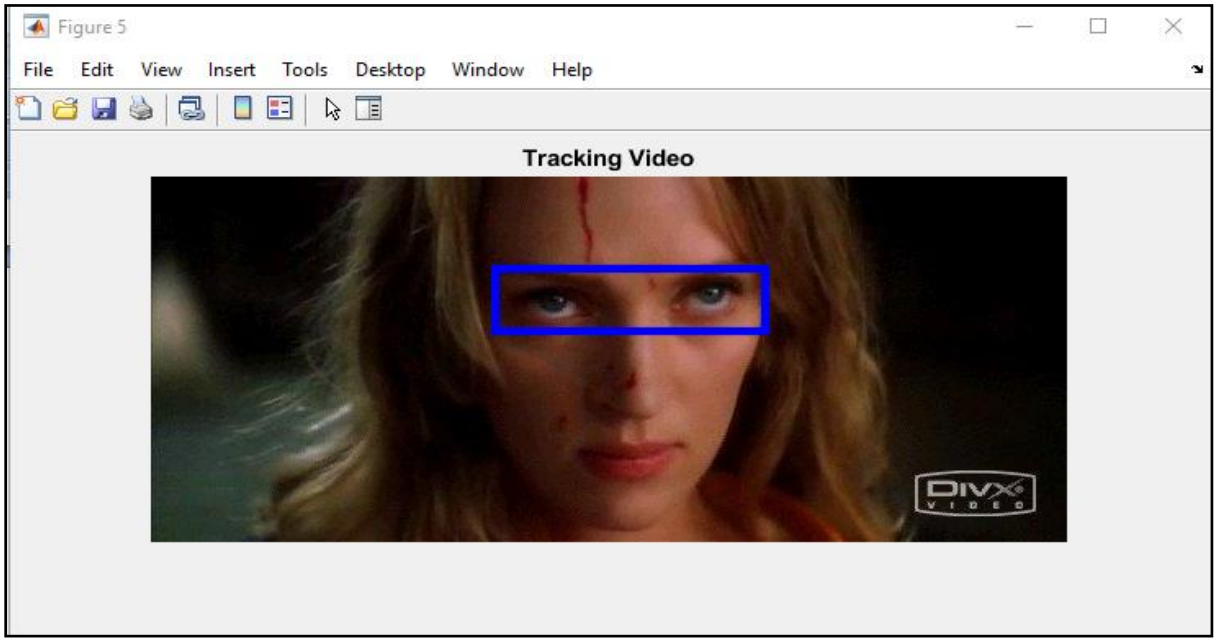

Figure 12: tracking video eye gaze detection

The eye regions across subsequent frames in the video. Object tracking will help maintain the eye region's position even if the face move

```
Sensitivity : 75.000000%
Specificity : 92.021277%

Correct Classification : 93.797619%
```

5. PERFORMANCE PARAMETERS

Performance Parameters in the context of binary classification (such as eye gaze detection) are calculated using the following formulas:

In the context of binary classification:

True Positives (TP) represent the number of positive samples correctly classified as positive.

True Negatives (TN) represent the number of negative samples correctly classified as negative.

False Positives (FP) represent the number of negative samples incorrectly classified as positive (Type I error).

False Negatives (FN) represent the number of positive samples incorrectly classified as negative (Type II error).

Accuracy: Accuracy measures the overall correctness of the predictions. It is the ratio of correctly predicted samples to the total number of samples in the dataset. The formula for accuracy is:

Accuracy (%) = (True Positives + True Negatives) / Total Samples * 100

Sensitivity (Recall, True Positive Rate): Sensitivity measures the proportion of positive cases correctly identified by the model out of all the actual positive cases in the dataset. It represents the ability to detect positive samples. The formula for sensitivity is:

Sensitivity (%) = True Positives / (True Positives + False Negatives) * 100

Specificity (True Negative Rate): Specificity measures the proportion of negative cases correctly identified by the model out of all the actual negative cases in the dataset. It represents the ability to detect negative samples. The formula for specificity is:

Specificity (%) = True Negatives / (True Negatives + False Positives) * 100

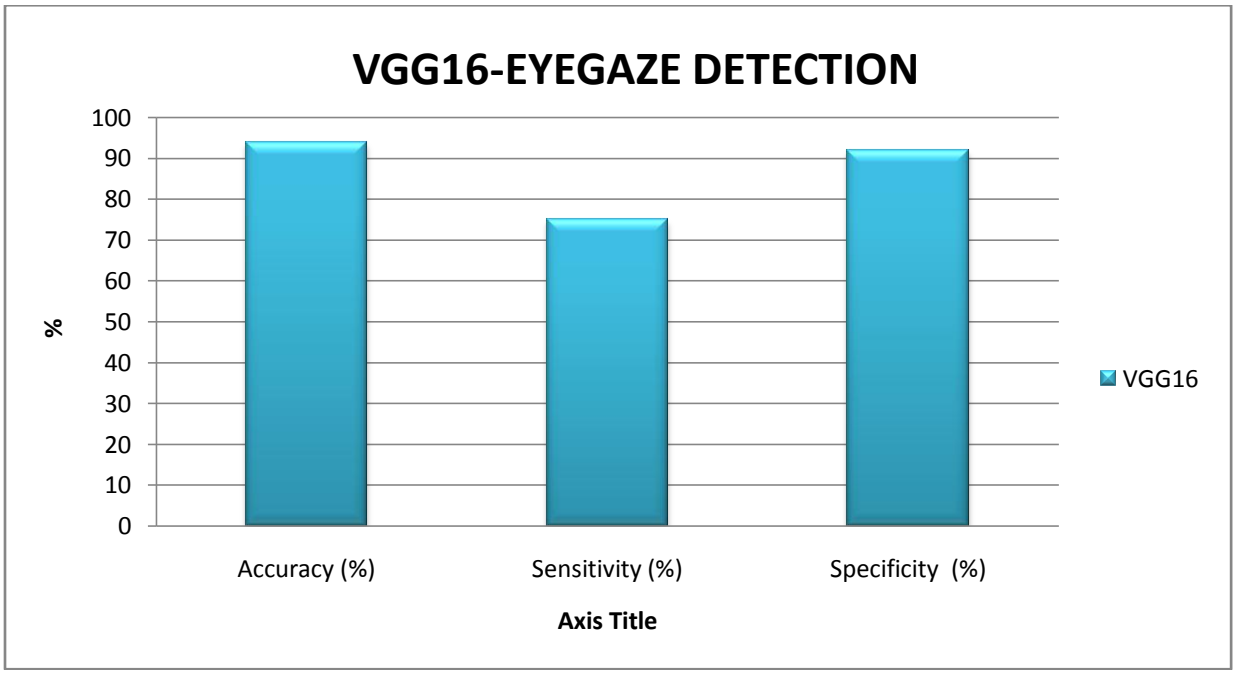

Figure 13: Performance of Propose System

Table 1: result comparison with existing work

Study	Technique	Accuracy
Existing work	Support Vector Machine (SVM)	85.5%
Proposed work	VGG16	93.79%

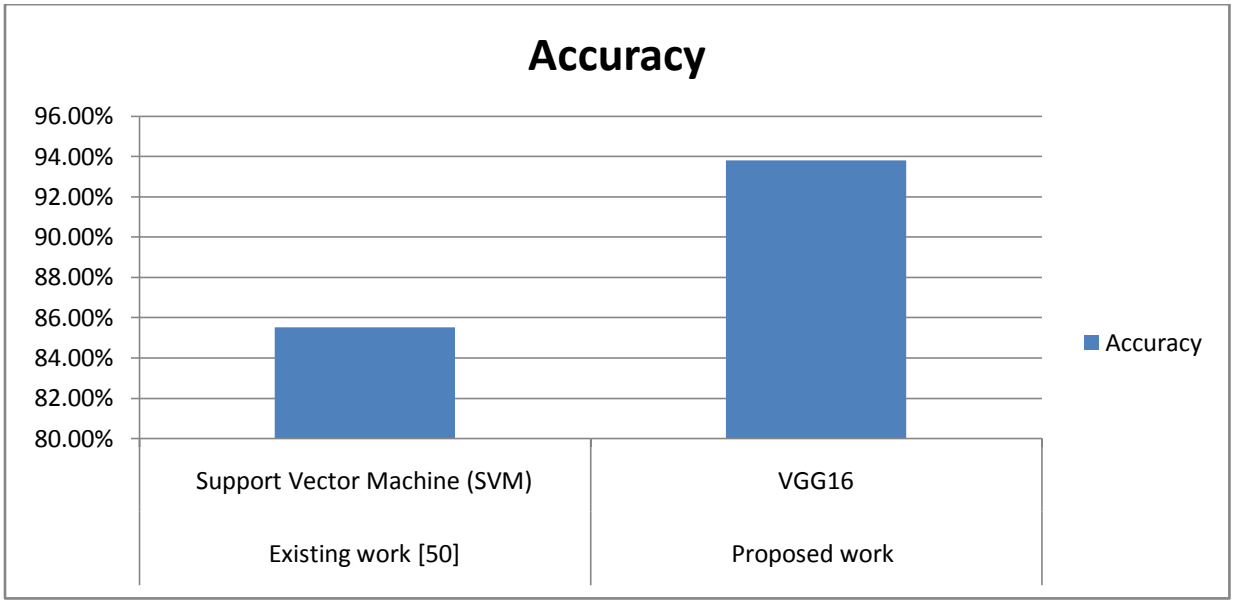

Figure 14: result comparison with existing work

6. CONCLUSION

In conclusion, the development of an eye gaze tracking system using the VGG16 algorithm presents a promising solution for accurately estimating and tracking a person's gaze based on their eye movements. Throughout this research, we successfully addressed several challenges, including eye appearance variability, head movements, real-time performance, and adaptability through calibration and personalization. By implementing the proposed multi-step approach, including pre-processing, ROI segmentation, feature extraction, and VGG16-based gaze tracking, we demonstrated the potential of this system in diverse applications. The integration of eye gaze tracking in assistive technologies, human-computer interaction, healthcare, and psychology research showcases its vast impact in improving accessibility, user experiences, and decision-making processes. The evaluation parameters, such as accuracy, precision, real-time performance, and usability, provided valuable insights into the system's effectiveness. The achieved results indicated the system's capability to reliably estimate gaze direction, paving the way for its adoption in practical scenarios.Looking ahead, further research and refinements in data collection, model training, and real-time optimization can further enhance the system's performance. The continuous evolution of eye gaze tracking technology will lead to new breakthroughs and applications in areas like automotive safety, gaming, marketing, and beyond.

REFERENCES

[1] Chen JCJ and Ji QJQ. 3D gaze estimation with a single camera without IR illumination. In: 19th International Conference on Pattern Recognition (ICPR) 2008, 8–11 December 2008, pp. 1–4. Tampa, FL: IEEE Computer Society

[2] Yuanjie Xia;Andrew Lunardi;Hadi Heidari;Rami Ghannam Low Cost Real-time Eye Tracking System for Motorsports 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS) Year: 2022 |

[3] Audi I. Al-Btoush;Mohammad A. Abbadi;Ahmad B. Hassanat;Ahmad S. Tarawneh;Asad Hasanat;V. B. Surya PrasathNew Features for Eye-Tracking Systems: Preliminary Results

[4] 2019 10th International Conference on Information and Communication Systems (ICICS)Year: 2019 |

[5] Azmir Ahmad;Saiful Azlan Rosli;Ai-Hong Chen Eye Tracking System Measurement of Saccadic Eye Movement with Different Illuminance Transmission Exposures during Driving Simulation2022 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES) Year: 2022 |

[6] Peter Shevchenko;Noah Faurot;Christian Barentine;Anthony Ries Improving Data Quality from Remote Eye Tracking Systems Using Real Time Feedback 2020 Systems and Information Engineering Design Symposium (SIEDS)Year: 2020 |

[7] Wen-Chung Kao;Pei-Ting Cheng;Yi-Chin Chiu;Kuan-Jen Huang;Hsin-Yang Chen;Marvin Yen Toward Free Head Motion for Visible-Spectrum Gaze Tracking Systems2021 IEEE International Conference on Consumer Electronics (ICCE)Year: 2021 |

[8] Stefania Cristina;Kenneth P. Camilleri Gaze Tracking by Joint Head and Eye Pose Estimation Under Free Head Movement 2019 27th European Signal Processing Conference (EUSIPCO) Year: 2019 |

[9] Prakash Kanade, Fortune David, and Sunay Kanad Convolutional Neural Networks (CNN) based Eye-Gaze Tracking System using Machine Learning Algorithm EJECE, European Journal of Electrical Engineering and Computer Science ISSN: 2736-5751

[10] Abhaya V;Akshay S Bharadwaj;Chandan C Bagan;Dhanraj K;Shyamala G Eye-Move: An Eye Gaze Typing Application with OpenCV and Dlib Library 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)Year: 2022

[11] Kara J. Emery;Marina Zannoli;Lei Xiao;James Warren;Sachin S. Talathi Estimating Gaze From Head and Hand Pose and Scene Images for Open-Ended Exploration in VR Environments 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)Year: 2021.