



Article

A Robust Framework for Real-Time Iris Landmarks Detection Using Deep Learning

Muhammad Adnan ¹, Muhammad Sardaraz ¹, Muhammad Tahir ^{1,*}, Muhammad Najam Dar ²,
Mona Alduailij ³ and Mai Alduailij ³

¹ Department of Computer Science, Attock Campus, COMSATS University Islamabad, Attock 43600, Pakistan; fa18-rs-040@cuiatk.edu.pk (M.A.); sardaraz@cuiatk.edu.pk (M.S.)

² Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad 46000, Pakistan; najam.dar77@ce.ceme.edu.pk

³ Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia; maalduailej@pnu.edu.sa (M.A.); maalduailej@pnu.edu.sa (M.A.)

* Correspondence: m_tahir@cuiatk.edu.pk

Abstract: Iris detection and tracking plays a vital role in human–computer interaction and has become an emerging field for researchers in the last two decades. Typical applications such as virtual reality, augmented reality, gaze detection for customer behavior, controlling computers, and handheld embedded devices need accurate and precise detection of iris landmarks. A significant improvement has been made so far in iris detection and tracking. However, iris landmarks detection in real-time with high accuracy is still a challenge and a computationally expensive task. This is also accompanied with the lack of a publicly available dataset of annotated iris landmarks. This article presents a benchmark dataset and a robust framework for the localization of key landmark points to extract the iris with better accuracy. A number of training sessions have been conducted for MobileNetV2, ResNet50, VGG16, and VGG19 over an iris landmarks dataset, and ImageNet weights are used for model initialization. The Mean Absolute Error (MAE), model loss, and model size are measured to evaluate and validate the proposed model. Results analyses show that the proposed model outperforms other methods on selected parameters. The MAEs of MobileNetV2, ResNet50, VGG16, and VGG19 are 0.60, 0.33, 0.35, and 0.34; the average decrease in size is 60%, and the average reduction in response time is 75% compared to the other models. We collected the images of eyes and annotated them with the help of the proposed algorithm. The generated dataset has been made publicly available for research purposes. The contribution of this research is a model with a more diminutive size and the real-time and accurate prediction of iris landmarks, along with the provided dataset of iris landmark annotations.

Keywords: iris; landmarks; deep learning; human–computer interaction



Citation: Adnan, M.; Sardaraz, M.; Tahir, M.; Dar, M.N.; Alduailij, M.; Alduailij, M. A Robust Framework for Real-Time Iris Landmarks Detection Using Deep Learning. *Appl. Sci.* **2022**, *12*, 5700. <https://doi.org/10.3390/app12115700>

Academic Editor: Byung-Gyu Kim

Received: 5 April 2022

Accepted: 2 June 2022

Published: 3 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Iris landmark localization is a significant element applied in various fields, including security, health, augmented reality, computer vision, and many more [1]. The main focus and the ultimate goal of iris landmark localization is to localize the iris key landmark points to extract irises with higher accuracy [2]. The iris knowledge gained from iris landmark locations can be handy in many recognition systems such as human–computer interaction, identification, and security programs [3]. Iris detection plays an essential role in different fields such as automatic biometric systems, custom clearance, and identity matching by police and can be very effective in identifying a citizen. Iris detection and tracking-based applications are more stable compared to fingerprints as fingerprints disappear with time or due to some diseases. There is no need to contact the machine, and the machine can easily take images of a walking person. Most importantly, environmental and bodily changes do not affect the iris [4]. However, iris detection is a time-consuming and computationally

expensive problem [5]. Human–computer interaction, remote tracking, enjoyment, security surveillance, and medical applications can be facilitated by facial information from facial landmark areas [6].

Iris landmark detection and tracking are still open areas of research. With iris landmark detection and iris tracking, a mobile device can be used, which is beneficial for disabled individuals. We provide a manually annotated iris landmarks dataset that can help researchers to localize and track the iris. We also provide a comparison between deep learning techniques to find a technique that performs better. The previous studies have worked on techniques that do not need any datasets because they use some image segmentation techniques to detect the iris, and these techniques do not require a dataset. The previous studies have used the old methods for iris detection, such as the Hough transform and contour detection, to detect the iris; however, these methods also need very high-quality images to detect iris. However, MobileNet version 2 is deep learning-based—an updated version—reducing the model size and increasing the response rate compared to the other models used in real-time scenarios.

This article proposes a robust framework for iris key landmark localization to extract the iris efficiently. It is a viable approach for aiding handicapped people to use a mouse using their iris. A dual-shot detector is used for face detection, and the ShuffleNet.v2 model is used to identify facial landmarks [7–9]. This research demonstrated the effectiveness of various Convolutional Neural Network (CNN) models on performance parameters selected to evaluate the proposed model. The main contributions of this research are as follows:

- (a) We collected a novel annotated dataset of iris landmarks and made it publicly available;
- (b) We proposed a deep learning-based model for the robust and real-time identification of iris landmarks.

The rest of the article is organized as follows. Section 2 discusses the related state-of-the-art research for iris landmark localization and applications, while Section 3 provides a detailed description of the acquisition of our novel dataset of iris landmarks and proposed methods. Section 4 is dedicated to the results and discussion, while conclusive remarks are provided in Section 5.

2. Related Work

Several applications such as authorization, identification, and surveillance are based on iris recognition systems. Near-infrared (NIR) [10] sensors are used for iris identification and extracting iris patterns containing enough textural information regardless of the color of the eyes.

A SoftMax classifier and CNN-based approach is presented in [11] to extract the discriminative features from the input images to locate the iris. The purpose of iris recognition is to develop a variety of user authentication and security applications. A high-quality camera was used to take images that needed to be very close to the eyes, and these were used for iris localization. Iris recognition is one of the best and most widely used biometric authentication methods and is more secure than the other methods. These have been used in large-scale applications. However, it is necessary to identify the originality of the iris to improve security. In some industries, contact lenses are used, which reduces the accuracy of the iris recognition system.

Several studies on iris tracking and iris detection exist in the literature. This section presents details of the state-of-the-art techniques, methods, datasets, applications, and performance of the prior studies. The authors in [12] used a dataset containing 66 pupil videos with exceptional quality that allowed pupil detection and the evaluation of existing methods. According to the results, the detection rate of half of all the cases is insufficient, and the dataset needs to be enhanced to support the best solution for pupil detection. An eye-tracking portable device with a high-quality camera was used to capture the pupil videos in different environmental situations. The pupil detection performance in outdoor and indoor scenes varies due to other environmental conditions. Computer vision has faced many problems earlier, and image segmentation was one of them, playing a critical

role in several real-world scenarios. The eye segmentation dataset presented in [13] is based on low-resolution images. Some landmark models, i.e., Active Appearance Model (AAM), Ensemble Regression Tree (ERT), and deep semantic segmentation models such as Atrous Convolutional Neural Network (ACNN), are implemented to evaluate the performance of the dataset. ACNN performs well in terms of IOU-based ROC curves.

We cannot negate the significance of iris segmentation for detection and recognition. More precise segmentation supports more accurate iris recognition. The authors in [14] proposed a unique model that implements U-NET with some fully enlarged or dilated convolutions, named the improved U-NET model. CASIA-iris-interval-v4.0, ND-IRIS-0405, and UBIRIS.v2 were used to evaluate the performance, and the f1-scores of the three datasets stated above were 97.36%, 96.74%, and 94.81%, respectively. Abate et al. [15] proposed a technique for iris detection centered on the watershed, and it primarily works with mobile devices. The watershed helps to estimate the boundary of the iris more precisely. The watershed is an image processing technique used to enhance the edges in the images, and in this work, this technique was used for iris boundary detection. After that, the circle fitting was applied to the iris boundary to find the border of the pupil. This method was analyzed on 15,000 eye images of 75 individuals. Gaze estimation plays a vital role in the era of interactive systems. A piece of helpful information perceived from the pupil, iris, and corners of the eyes is enough to estimate the gaze directions. Iris localization is an initial and essential setup in gaze tracking. Sheela et al. [16] proposed a method that uses cascade classifiers for eye and face detection, Haar features for the pupil, sclera, eye corners detection, and the Hough gradient for iris localization. According to the different existing studies, the performance of cascade classifiers is not suitable for face and eye detection. Template matching techniques are used to estimate the variation in the frames, which enhances the performance. Biometrics necessitates recognizing a human being based on their behavioral and biological traits. Initially, image segmentation, localization, and feature extraction were performed. Then, a 1D-Gober filter was used to extract distinctive characteristics of the iris, and humming distance was involved in matching the iris images. The evaluation of the proposed model was performed on UBIRIS v1, IIT DELHI DATABASE, and CASIA-IRIS TWINS datasets. The enhancement of the database may improve the performance of the proposed model [17].

A Deep Learning (DL) and Gray-Level Co-Occurrence Matrix (GLCM) based methodology is proposed in [18] for iris liveliness detection and localization. Energy, entropy, contrast, and correlation features were obtained with the help of GLCM. Several deep learning models such as VGG16, VGG19, DenseNet121, DenseNet169, DenseNet201, and ResNet50 were applied to Clarkson LivDet2013, Clarkson LivDet2015, and IIITD datasets to evaluate the performance of the model. Eye movements play an imperative role in eye tracking, and they also contribute to human–computer interaction. A pre-trained convolution neural network-based model was implemented to segment the iris by Aayush et al. [19]. The Speed Up Robust Features (SURF) feature was also used to feature vectors. Experiments were conducted on costly hardware such as a high-quality camera, and to illuminate cheek regions, Infrared IREDs were utilized.

Iris localization is a significant step in iris identification systems, and its correctness may directly affect iris normalization, feature extraction, and matching steps. The authors in [20] proposed a Coarse-To-Fine (CTF) algorithm-based computational cost-effective method, and also Daugman's integrodifferential operator was used to refine the pupil and iris centers. The proposed technique was evaluated on the CASIA-V3-Interval database of images.

Different computer techniques are helpful in producing segmented eye parts; later on, these can be used in iris tracking-based video oculo-graphy. A convolutional neural network-based approach was proposed by Kothari et al. [21], named EllSeg. The prediction of complete elliptical eye regions with the background was made through the EllSeg framework. The presented model was trained and evaluated with NVGaze, OpenEDS, RITEyes, ElSe, ExCuSe, PupilNet, and LPW datasets with significantly less error margin.

Fast iris position tracking is essential when working with real-time applications to assist human–computer interaction. However, in most eye image databases, the irises in the images are affected by different noises. Ma et al. [22] proposed a novel Conformal Geometric Algebra (CGA) based algorithm which is used to track the iris position more precisely. Initially, the grayscale images were converted into three channels images with an adaptive threshold, and a Sobel edge detector was implemented to produce edge points. Finally, an enhanced CGA-based algorithm was used to detect circles for iris boundaries estimation.

Properly extracting the iris areas from images is the critical step in the identification of an individual. Therefore, we cannot deny the importance of segmentation in the localization of the iris. The authors in [23] came up with a novel technique that employs rough entropy, a soft-computing approach to segment the iris, and Circle Sector Analysis (CSA) was used to measure the iris error-free position. The proposed technique was compared with the circular Hough transform, which is a state-of-the-art (iris region segmentation) technique, on the CASIA-Iris-V3-Interval, IITD, and MMU1s datasets. According to the authors, the presented work performed well compared to the state-of-the-art methods. However, this method is still computationally expensive to execute in real-time scenarios.

The use of cameras in smartphones can provide benefits to disabled persons with gaze estimation. Park et al. [24] proposed a stacked-hourglass network-based learning method for eye region landmarks localization. EYEDIAP, MPIIGaze, UT Multiview, and Columbia Gaze datasets were used in evaluating the proposed model. Real-time gaze estimation applications can implement these types of models to aid human–computer interactions.

Iris recognition systems need efficient algorithms or models that have less computational cost and that can be used in security-sensitive applications in real time. Haar cascade classifiers and CNN-based techniques were developed to recognize head movements to control the cursor, and different Haar features such as the edge-line and four rectangles were used [25].

In [26], the authors proposed a hybrid approach that uses the circular Hough transform, Viola–Jones, and Kanade–Lucas–Tomasi (KLT) algorithms for eyeball tracking. The Hough transform is an image processing approach for detecting circles in distorted images. The Hough transform technique combined with a Canny edge detector was used for pupil detection. Additionally, this study demonstrated the hardware required for the proposed method. They claimed that the proposed model achieved 96% accuracy.

In [27], higher efficiency is claimed when using the combination of double threshold, morphological operations, and center of gravity method. It also required a very high-quality image to proceed further with pupil detection in iris images. This technique requires specific hardware such as a camera placed very close to the eyes for image acquisition purposes. The main drawback is that it cannot be used in real-time; if the author tries to apply this in real-time, they need to make an expensive but tiny device containing a very small camera with excellent resolution. Recently, several significant developments have been made in image processing disciplines, mainly in feature identification methods, and researchers are struggling to develop accurate and effective feature detection algorithms. Eye detection has been achieved using a feature-based approach that uses low-quality attributes such as corners, color, form, and texture [28].

Eye movement is an appropriate communication medium between humans and computers, particularly helpful for disabled persons. Template, appearance, and feature-based methods are also eye detection techniques with great accuracy but are computationally expensive. In the 1990s, researchers conducted research in which they adopted a template-based approach for eyes detection. Initially, a standard eye-shaped template is crafted, and it is time-consuming, requires high-quality eye photos, and is restricted to frontal view images [29]. According to the article [30], the head-mounted devices are essentially based on a customized structure that keeps a small camera in front of the user’s face. This mini camera helps capture the high-quality and detailed images of the human eye used for accurate pupil detection. However, this study demands a high-quality camera embedded in a particular device, which may be costly.

Iris landmarks detection and tracking are still open areas of research. With iris landmarks detection and iris tracking, we can easily use a mobile device with the iris, which will be very beneficial for disabled individuals. The previous studies have been working on techniques that do not need a dataset because they are using some image segmentation techniques to detect the iris, and also these techniques do not require a dataset. As the previous studies have been using the old methods for iris detection, such as the Hough transform and contour detection, to detect the iris, these methods also need very high-quality images to detect the iris. However, MobileNet version 2 is deep learning-based—an updated version—reducing the model size and increasing the response rate compared to the other models when used in real-time scenarios.

3. Materials and Methods

This section presents details of the proposed model. First, details of the datasets are presented followed by detailed steps of the proposed model.

3.1. Dataset Description

Iris landmark localization is a significant element applied in various fields, including security, health, computer vision, and many more. An iris key landmarks dataset is not publicly available. Therefore, a synthetic dataset was created from face images and videos. We used two self-recorded and three downloaded videos from YouTube. All videos are face-focused because extracting the eye image is necessary. We recorded the videos with a Samsung mobile phone with a 13 Megapixel camera. All the recorded videos have good quality.

Firstly, we checked whether the videos were face-focused and identified the irrelevant parts that were not used to create the dataset. Different video editing tools that are publicly available were used to remove the different parts of the video that were not used in the creation of the dataset. The details of these videos are shown in Table 1.

Table 1. Details of dataset sources.

Source	Gender	Size MBs	Resolution	Frames Per s	Duration (s)	Total Images	Included Images
Self Recorded	Male	57	1920 × 1080	30.1	54	1625	1500
Youtube	Female	16	1280 × 720	25	324	8100	5690
Self Recorded	Male	34	1920 × 1080	30.1	32	963	700
Youtube	Female	26	3840 × 2160	25	20	500	453
Youtube	Female	27	1920 × 1072	24	290	6960	5050

The media pipe [31] is a powerful library for face and facial landmarks detection. The library is used to extract eye images. The process starts by reading all the videos frame by frame and passing each frame to the library for facial landmarks detection. Images are saved for the left and right eye separately.

The next step is to clean the data to achieve better images that will help us to train and test the model accurately. The images are checked to remove the images containing a closed eye because if the iris is not visible in any image, annotation and predictions cannot be performed on the images. The details of the datasets are presented in Table 1.

3.2. Data Acquisition

An iris key landmarks dataset is not publicly available. Therefore, a synthetic dataset was created from face images and videos. Initially, the videos were converted into images to extract high-resolution photos of the eyes. Five key landmarks points of the iris were marked to retrieve coordinates of iris landmarks, i.e., center, top, right, bottom, and left, as shown in Figure 1. The total number of the annotated iris landmarks images is 13,393.



Figure 1. Iris landmark annotation samples.

3.3. Iris Landmark Annotations

To annotate the iris, five key landmarks i.e., center, top, right, bottom, and left are identified. To handle the problem of the different diameters of the irises of different persons, the threshold method was used to check that if the iris diameter and the area selected by the proposed method were the same. After this, the annotation was started. The radius was adjusted to obtain a perfectly aligned circle with the iris. The threshold value was added or subtracted from the center of the iris. Algorithm 1 shows the steps involved in landmarks annotation.

Algorithm 1: Iris landmark annotation pseudo code.

```

Input : Mouse_Event, x,y, file_name
Output: Create a text file with the same image name and save the annotations in
          that file
Procedure Annotate_Image
if (Mouse_Event is equal to Mouse Move then
  | Draw a circle to check whether Matches Iris diameter or not
end
if Mouse_Event is equal to Mouse Left Click then
  | Draw Circle on center, Draw Circle on Top, Draw Circle on Right, Draw Circle
  | on Bottom, and Draw Circle in the left side of the iris.
end
if Mouse_Event is equal to Mouse Right Click then
  | generate output
end
Procedure Main
Create a directory to save images with annotations
Input : Initialize a list with Image Names
while images are available do
  | read Images call: Annotate_Image display image till next image or program
  | ends
end

```

3.4. Proposed Methodology

Many efficient neural network architectures are based on Depthwise Separable Convolutions (DSC). The main idea of DSC is to substitute the full convolutional layers into depth-wise and point-wise convolutions. Point-wise convolutions are also known as 1×1 convolutions. This substitution is made through the factorization of a standard convolution. The primary objective of the depth-wise layer is to apply a single lightweight filter to every given input channel. Point-wise convolutions are responsible for computing the combination of input channels in a linear way. A feature map represented as F with dimensions $D_F \times D_F \times M$ is fed as an input to the standard convolutional layer, and a convolutional kernel K of size $D_K \times D_K \times M \times N$ is applied. The computational cost of a standard convolution is calculated as shown in Equation (1).

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{1}$$

when the convolutional kernel K is applied, it produces a feature map G with the $D_F \times D_F \times M$ dimensions, where input and output channels are represented by M and N , respectively. The feature map G for standard convolution is computed as stated in Equation (2).

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \tag{2}$$

The computational complexity depends on the multiplication of the input, output channels, kernel size, and feature map, which are $M, N, D_K \times D_K$ and $D_F \times D_F$, respectively. MobileNet models express each of these components and their relationship. The model uses DSC to split the relationship between the kernel size and the number of output channels. Only one filter is implemented to each input channel via DSC. The MobileNetV2 architecture comprises 32 filters and 19 residual bottleneck layers. As a non-linearity, ReLU6 is preferred due to its reliability [32]. MobileNet always uses batch normalization, dropout, and the standard kernel size for modern 3×3 networks during the training process. One filter is implemented for each input channel of DSC, and the resulting feature map \hat{G} is given in Equation (3).

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \tag{3}$$

$D_K \times D_K \times M$ is the size of the depth-wise convolutional kernel \hat{K} and the m_{th} filter in the kernel \hat{K} . It is employed for the m_{th} channel in feature map F to determine the n_{th} channel of the filtered output feature map \hat{G} . The computational complexity of DSC is shown in Equation (4).

$$D_K \times D_K \times M \times D_F \times D_F \tag{4}$$

The term DSC refers to the combination of 1×1 (point-wise) and depth-wise convolution, as presented in [33]. Equation (5) shows the computational complexity of DSC.

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \tag{5}$$

This computational complexity involves adding the depth-wise and 1×1 point-wise convolution, as presented in Equation (6).

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{6}$$

The base of the MobileNet architecture is very light in size and has low latency. In many real-world scenarios, different applications demand lightweight and efficient models. In [34], the authors proposed two new parameter: the width multiplier, represented as α , and the resolution multiplier. The width multiplier aims to reduce a network at each layer uniformly, and the resolution multiplier ρ is employed to the input image, so the internal presentation of each layer is consequently decreased. α, ρ, M, N are the symbols

of the width multiplier, resolution multiplier, input channels, and the output channel, respectively. Both width and resolution multipliers are applied, and the result is provided in Equation (7).

$$D_K \times D_K \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F \tag{7}$$

where $\rho \leq 1, \alpha \leq 1$, and $\alpha, \rho \in (0, 1)$. With the help of the width multiplier and resolution multiplier, the computational complexity is reduced by α^2 and ρ^2 , respectively.

3.4.1. Proposed Model

Figure 2 shows the flow diagram of the proposed framework. An image is sent to the face detection model to predict the face bounding box after it has been rescaled and normalized as required. After scaling and normalizing, this prediction is provided as an input to the facial landmarks model to predict 68 facial landmarks. Finally, we acquire eye-cropped images from the landmarks model. These eye images are rescaled and normalized. The iris landmarks model is then fed to the eye-cropped images for prediction. A complete pipeline of the face bounding boxes, facial landmarks, and iris landmarks models is given in Figure 3.

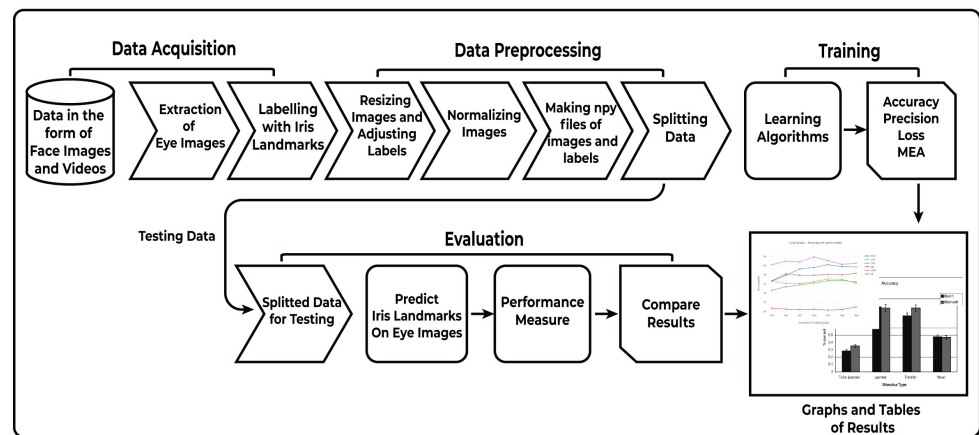


Figure 2. The flow of the proposed model.

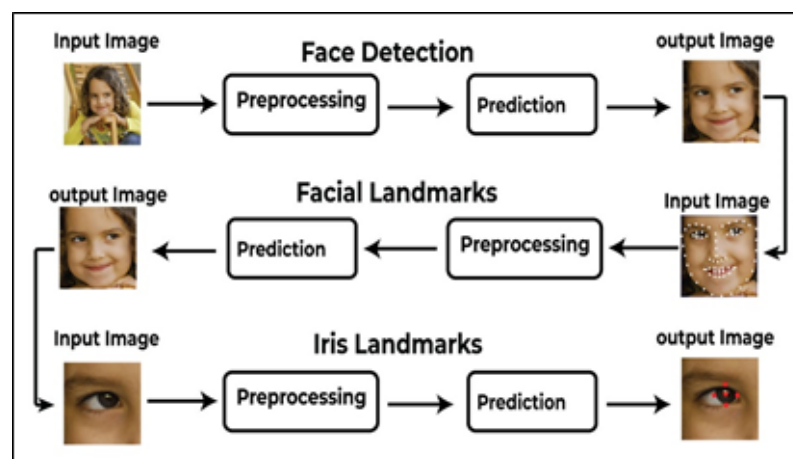


Figure 3. Iris detection steps of the proposed model.

3.4.2. Preprocessing

Data normalization is the process of assigning the same value to each input parameter or pixel. Because each input pixel in the image data has a value between 0 and 255, it is highly advisable to normalize the pixel values to values between 0 and 1. We trained our models with 64×64 input shapes; resizing the images is essential to help the models

during the training process. We needed to adjust the x, y coordinated values of labels according to the resized image to match the new dimensions of the image, working with NumPy files that simplify and accelerate the development of the computer vision and machine learning models. Thus, we converted all image data to npy files and all label data to a separate file. We needed to segment the dataset for training and testing. In this research, we segmented the data with a 90/10 ratio into training/testing sets.

3.4.3. Training and Testing

The training phase involved two stages. The first was feeding data into the model, known as the learning or training stage. The second was testing the model through training and obtaining results for various parameters such as accuracy, MAE, and loss. We conducted training sessions using the iris landmarks dataset for MobileNetV2 [35], ResNet50 [23], VGG16 [24], and VGG19 [8], and these models were initialized with ImageNet weights [25]. The training of these models was performed over 300 epochs using the Adam optimization technique [26] with a learning rate of 0.001, batch size of 64, and MAE loss function [27]. A complete flow diagram is shown in Figure 4.

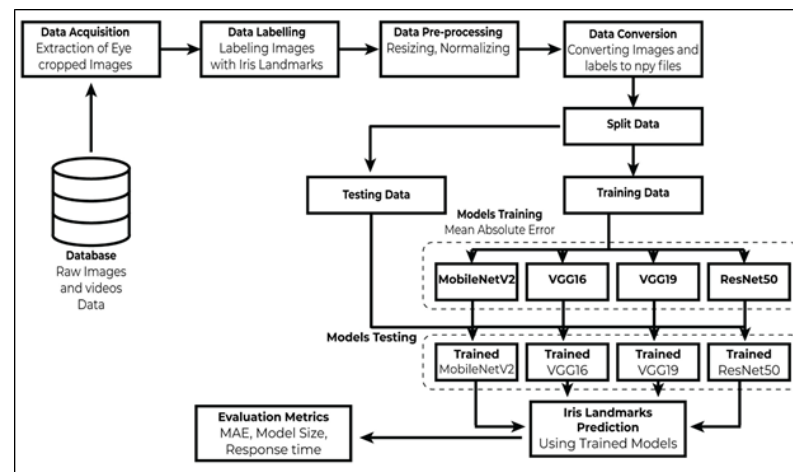


Figure 4. Training and testing procedure of proposed model.

In this research, we used four different deep learning models, i.e., MobileNet version 2, VGG16, VGG19, and ResNet50, over a self-collected and annotated iris landmarks dataset. All models had the same configurations: i.e., the number of layers, optimizers, learning rate, etc. ImageNet weights were used to initialize all the models. For all models, the input shape of 64×64 was used. All images passed as input to each model had three channels. This is the reason behind the use of three channels for the input shape. Different layers were added such as GlobalAveragePooling2D layers, and the dropout rate was set to 0.3. The last dense layer was added with the value of 10. Adam was implemented as the optimizer for all models, and the mean square error (MSE) was used as a loss function. The learning rate was set to 1×10^{-6} . All checkpoints of each model were saved at a specific interval, which was helpful for examining the model performance and can be used in the future. The training of these models was performed over 300 epochs with a learning rate of 0.001 and batch size of 64. The details of the number of parameters selected for each parameter are shown in Table 2.

Table 2. Number of parameters used for each model .

Model	Number of Parameters	Trainable Parameters	Non-Trainable Parameters
MobileNet V2	2,270,794	2,236,682	34,112
ResNet50	23,608,202	23,555,082	53,120
VGG16	14,719,818	14,719,818	0
VGG19	20,029,514	20,029,514	0

4. Results and Discussion

This section presents the results and discussion. The experimental setup is discussed, followed by the detailed results and discussion.

4.1. Experimental Setup

The proposed model was implemented in Python 3.7 with NumPy, OpenCV, TensorFlow [36], and Keras libraries. The models were implemented with the Keras framework with batch sizes of 64 and 300 epochs, we used the Adam Optimizer for optimization purposes, and the learning rate decreased according to the given threshold throughout the training process. We used an Nvidia GeForce GTX 1650 4 GB GPU for training the models.

4.2. Model Evaluation

Evaluation is a significant stage that uses the test data to predict the results and evaluate the models. After that, the test data are passed to the model. The model then performs some actions and utilizes the acquired knowledge to predict iris landmarks on the images during the learning phase. Graphs and tables are used to represent data. We can quickly evaluate the performance of any model using graphs, as each model's performance is represented in a single graph. Additionally, the table is essential for expressing the numerical and textual analysis and comparing the proposed model and existing literature studies.

Performance metrics are used to measure the performance of the model. Accuracy is the main statistical performance parameter primarily used for the evaluation of a model. If iris landmarks are detected, there are two possibilities: whether a landmark is true or not, a true positive (*TP*) and a false positive (*FP*) can be found. Similarly, if iris landmarks are not detected, there are also two possibilities: i.e., (*TN*) as true negative and (*FN*) as a false negative. The formula to measure the accuracy of a model is shown in Equation (8).

$$Accuracy (ACC) = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \quad (8)$$

The Mean Absolute Error (MAE) is a statistical measure of errors between the measured value and actual value. The formula for measuring MAE is shown in Equation (9). The model size is also a parameter used to compare the size of the model. Response Time (*RT*) is also an evaluation parameter. The formula for calculating RT is given in Equation (10).

$$MSE = \frac{1}{N} \sum_{i=0}^N |y - \hat{y}_i| \quad (9)$$

$$RT = sp_t - ep_t \quad (10)$$

where *sp* is starting prediction, *ep* is ending prediction, and *t* is the time for RT calculation.

4.3. Results

The iris is detected using the MobileNetV2, ResNet50, VGG16, and VGG19 models. The performance of the different models over different metrics, i.e., MAE, model size, loss, and response time, is presented. An example of the detection of an iris with the proposed model is shown in Figure 5. Iris landmarks for different images are marked.

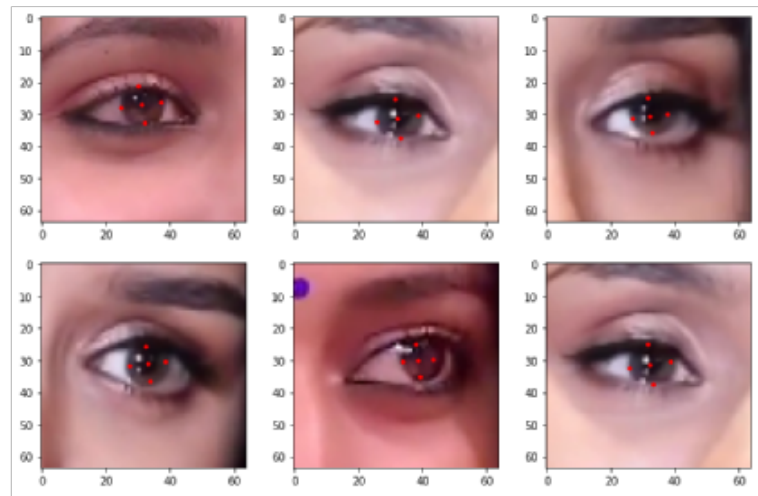


Figure 5. Prediction of iris landmarks of the proposed model.

4.3.1. Mean Absolute Error

MAE is used as a loss function and performance parameter in this study. According to the results in Figure 6, the MAE of ResNet50 is less than the other models. However, it cannot be concluded that this model performs very well because other parameters are also influencing the models. After the discussion on other parameters, the performance of all models over all parameters can be determined. The MAE comparison of all the models is given below in Figure 6.

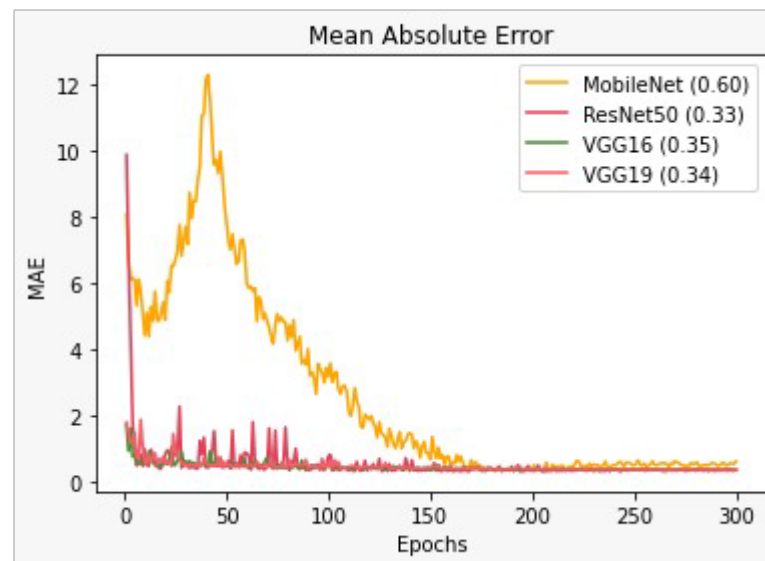


Figure 6. Mean Absolute Error of all the models.

4.3.2. Response Time

Response time is significant for real-time iris landmark detection applications. The results are presented in Table 3. The results show that MobileNet V2 is efficient compared to the other models.

Table 3. Response time of the models.

Model	Response Time (ms)
MobileNetV2	14.43
ResNet50	47.04
VGG16	108.38
VGG19	134.13

4.3.3. Model Loss

The loss of all models is approximately equal during the testing phase; however, VGG16 and VGG19 have more loss values during the training phase. Figure 7a–d shows the model losses of MobileNetV2, ResNet50, VGG16, and VGG19, respectively, plotted with 300 epochs.

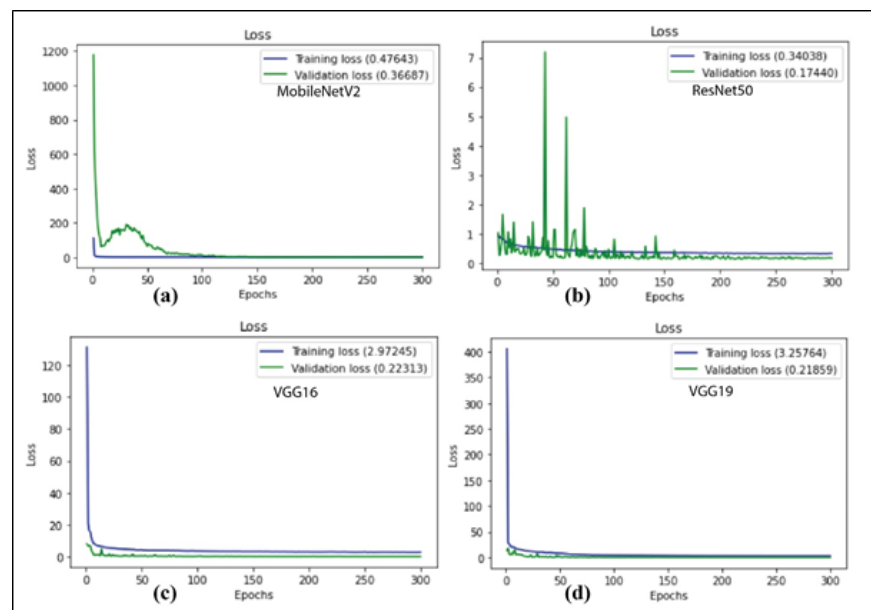


Figure 7. Model loss of iris key points detection over 300 epochs (a) MobileNetV2 (b) ResNet50 (c) VGG16 (d) VGG19.

4.3.4. Models Size

After training the models, we produced a tflite model of each network because these models must load readily and quickly when deployed in real time. When the model size is too large, it is challenging to load immediately. All other models except MobileNet are too large to use in real time. Figure 8 shows the size graph of each model.

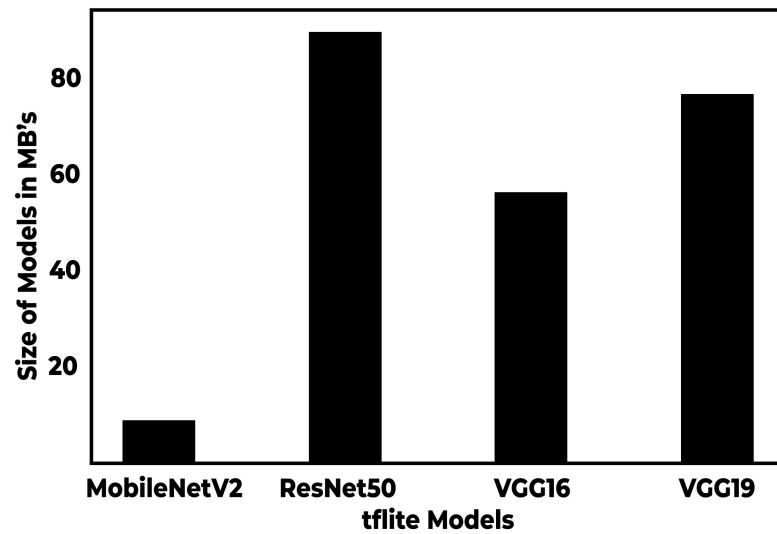


Figure 8. Size graph of each model.

4.3.5. Mean Intersection over Union

The Mean Intersection Over Union (mIoU), as presented in Figure 9, is an evaluation metric that is frequently used for semantic image segmentation. The Intersection Over Union, also known as the Jaccard index, is mainly used to measure the percentage of overlap between the prediction and actual output. The Dice coefficient is used in training as a loss function, and this is closely associated with this evaluation metric. The equation below presents a formula to calculate the Intersection over Union (IoU).

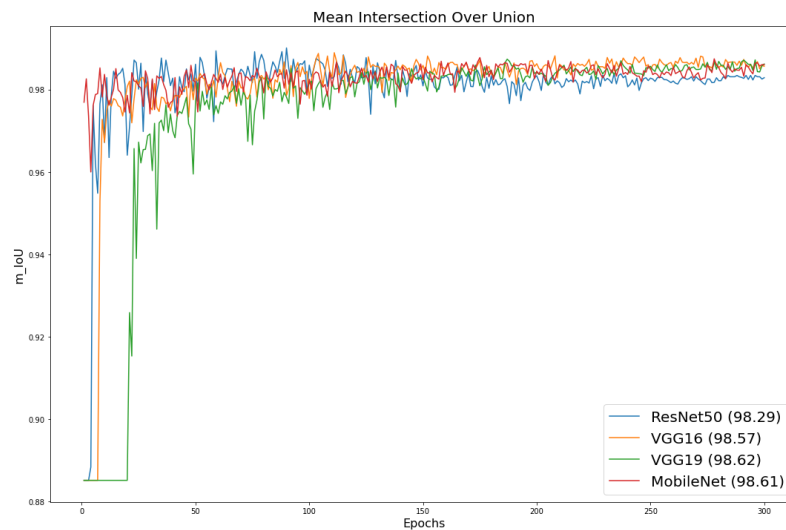


Figure 9. Mean Intersection Over Union of each model.

$$mIoU = \frac{1}{n} \sum_{c=1}^n TP / (TP + FP + FN) \quad (11)$$

First, with the help of the above formula, we calculated the IoU for all the classes and then computed the mean of all the values to provide a global value. Here, n represents the total number of classes, TP represents the true positives, FP represents the false positives, and FN represents the false negatives.

4.4. Discussion

The performances of all the models are good for different evaluation parameters such as MAE, Model Size, Response time, and mIoU (Mean Intersection over Union). The graph in Figure 6 shows that the performance of ResNet50 is better than other models in terms of Mean Absolute Error metrics. The response time of MobileNet version 2 is significantly less compared to the other models, as given in Table 3. The output size of MobileNet is more minimal than others. The most critical performance metric is the Mean Intersection Over Union (mIoU), which demonstrates that all the models' performances are good, but the MobileNet model performs outstandingly. After analyzing all the models on all performance metrics, we can say that the performance of MobileNet version 2 is exceptional over all the parameters as compared to ResNet50, VGG16, and VGG19.

5. Conclusions

This research presents a newly collected benchmark dataset of iris landmarks and makes it publicly available. A novel dataset that contains x and y coordinates of five iris-landmarks was created, which was not publicly available. In the preprocessing stage, we used image reshaping and normalization, which are helpful in model training. Various deep learning models were retrained for the regression problem of precisely predicting the iris landmarks, and we proposed the MobileNet v2 model-based robust framework. The robustness of the proposed framework lies in comparing various performance parameters such as Mean Absolute Error, model size, response time, validation loss, and Mean Intersection Over Union. The performance of MobileNet version 2 is exceptional regarding these parameters. Therefore, this research primarily provides an annotated benchmark dataset and validated the robustness of the MobileNet v2 model-based framework for the automatic, precise, and quick measurement of iris landmarks localization against the annotated dataset. This research contribution will allow researchers to compare their frameworks using this benchmark dataset against annotated iris landmarks for precise iris localization, which can be used in many applications such as iris base cursor movements to support a person with limb amputations.

Author Contributions: Conceptualization, M.A. (Muhammad Adnan) and M.T.; methodology, M.A. (Muhammad Adnan) and M.N.D.; software, M.A. (Muhammad Adnan); validation, M.A. (Mona Alduailij), M.S., M.A. (Mai Alduailij) and M.T.; formal analysis, M.S., M.T., M.A. (Mona Alduailij) and M.A. (Mai Alduailij); investigation, M.A. (Muhammad Adnan); resources, M.S., M.T.; data curation, M.N.D., M.T., M.S.; writing—original draft preparation, M.A. (Muhammad Adnan) and M.N.D.; writing—review and editing, M.S., M.T., M.A. (Mona Alduailij), M.A. (Mai Alduailij); visualization, M.T., M.A. (Mona Alduailij); supervision, M.T.; project administration, M.S., M.A. (Mai Alduailij). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Some datasets used are available publicly, and the datasets generated in this work are available at https://github.com/fa18-rcs-040/Iris_landmarks (accessed on 20 May 2022).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hussein, M.; Coats, D. Use of iris pattern recognition to evaluate ocular torsional changes associated with head tilt. *Ther. Adv. Ophthalmol.* **2018**, *10*, 2515841418806492. [CrossRef] [PubMed]
2. Lin, C.; Li, X.; Li, Z.; Hou, J. Finding Stars from Fireworks: Improving Non-Cooperative Iris Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2022**. [CrossRef]
3. Bruni, V.; Vitulano, D. A robust perception based method for iris tracking. *Pattern Recognit. Lett.* **2015**, *57*, 74–80. [CrossRef]
4. Jose, F.; Carvalho, S.; Manuel, J.; Tavares, R. Two methodologies for iris detection and location in face images. In *Computational Modelling of Objects Represented in Images*; CRC Press: Boca Raton, FL, USA, 2018; pp. 129–133.
5. Liu, Y.; Chen, C.; Zhang, M.; Li, J.; Xu, W. Joint Face Detection and Landmark Localization Based on an Extremely Lightweight Network. In Proceedings of the International Conference on Image and Graphics, Haikou, China, 6–8 August 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 351–361.

6. Ergen, B. Facial Landmark Based Region of Interest Localization for Deep Facial Expression Recognition. *Tehnički Vjesn.* **2022**, *29*, 38–44.
7. Li, J.; Wang, Y.; Wang, C.; Tai, Y.; Qian, J.; Yang, J.; Wang, C.; Li, J.; Huang, F. DSFD: dual shot face detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5060–5069.
8. Guo, X.; Li, S.; Yu, J.; Zhang, J.; Ma, J.; Ma, L.; Liu, W.; Ling, H. PFLD: A practical facial landmark detector. *arXiv* **2019**. arXiv:1902.10859
9. Ma, P.; Martinez, B.; Petridis, S.; Pantic, M. Towards practical lipreading with distilled and efficient models. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7608–7612.
10. Tripathi, P.; Akhter, Y.; Khurshid, M.; Lakra, A.; Keshari, R.; Vatsa, M.; Singh, R. MTCD: Cataract detection via near infrared eye images. *Comput. Vis. Image Underst.* **2022**, *214*, 103303. [\[CrossRef\]](#)
11. Zhang, X.; Liu, X.; Yuan, S.M.; Lin, S.F. Eye tracking based control system for natural human-computer interaction. *Comput. Intell. Neurosci.* **2017**, *2017*, 5739301. [\[CrossRef\]](#)
12. Tonsen, M.; Zhang, X.; Sugano, Y.; Bulling, A. Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, Charleston, SC, USA, 14–17 March 2016; pp. 139–142.
13. Luo, B.; Shen, J.; Wang, Y.; Pantic, M. The iBUG eye segmentation dataset. In Proceedings of the 2018 Imperial College Computing Student Workshop (ICCSW 2018), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, London, UK, 20–21 September 2018.
14. Zhang, W.; Lu, X.; Gu, Y.; Liu, Y.; Meng, X.; Li, J. A robust iris segmentation scheme based on improved U-net. *IEEE Access* **2019**, *7*, 85082–85089. [\[CrossRef\]](#)
15. Abate, A.F.; Frucci, M.; Galdi, C.; Riccio, D. BIRD: Watershed based iris detection for mobile devices. *Pattern Recognit. Lett.* **2015**, *57*, 43–51. [\[CrossRef\]](#)
16. Sheela, S.; Abhinand, P. Iris detection for gaze tracking using video frames. In Proceedings of the 2015 IEEE International Advance Computing Conference (IACC), Bangalore, India, 12–13 June 2015; pp. 629–633.
17. Khan, M.T.; Arora, D.; Shukla, S. Feature extraction through iris images using 1-D Gabor filter on different iris datasets. In Proceedings of the 2013 Sixth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2013; pp. 445–450.
18. Ramkumar, M.; Preeya, V.A.; Manikandan, R.; Karthikeyan, T. IIRIS detection for biometric pattern identification using deep learning. *Ictact J. Image Video Process.* **2021**, *12*, 2610–2614.
19. Chaudhary, A.K.; Pelz, J.B. Motion tracking of iris features to detect small eye movements. *J. Eye Mov. Res.* **2019**, *12*: 10.16910%2Fjemr.12.6.4. [\[CrossRef\]](#)
20. Soliman, N.F.; Mohamed, E.; Magdi, F.; Abd El-Samie, F.E.; AbdElnaby, M. Efficient iris localization and recognition. *Optik* **2017**, *140*, 469–475. [\[CrossRef\]](#)
21. Kothari, R.S.; Chaudhary, A.K.; Bailey, R.J.; Pelz, J.B.; Diaz, G.J. Ellseg: An ellipse segmentation framework for robust gaze tracking. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2757–2767. [\[CrossRef\]](#)
22. Ma, L.; Li, H.; Yu, K. Fast iris localization algorithm on noisy images based on conformal geometric algebra. *Digit. Signal Process.* **2020**, *100*, 102682. [\[CrossRef\]](#)
23. Sardar, M.; Mitra, S.; Shankar, B.U. Iris localization using rough entropy and CSA: A soft computing approach. *Appl. Soft Comput.* **2018**, *67*, 61–69. [\[CrossRef\]](#)
24. Park, S.; Zhang, X.; Bulling, A.; Hilliges, O. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In Proceedings of the 2018 ACM symposium on eye tracking research & applications, Stuttgart, Germany, 2–5 June 2020; pp. 1–10.
25. Lupu, C.; Turcu, C.O. Real-Time Iris Recognition of Individuals—an Entrepreneurial Approach. *EIRP Proc.* **2021**, *16*.
26. Dave, A.; Lekshmi, C.A. Eye-ball tracking system for motor-free control of mouse pointer. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 1043–1047.
27. Prathipa, R.; Premkannan, P.; ragunathan, K.; Venkatakrisnan, R. Human eye pupil detection technique using center of gravity method. *Int. Res. J. Eng. Technol.* **2020**, *7*, 3818–3821.
28. Vijayalaxmi, B.; Anuradha, C.; Sekaran, K.; Meqdad, M.N.; Kadry, S. Image processing based eye detection methods a theoretical review. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1189–1197. [\[CrossRef\]](#)
29. Archana, M.; Nitish, C.; Harikumar, S. Real time Face Detection and Optimal Face Mapping for Online Classes. *J. Phys. Conf. Ser.* **2022**, *2161*, 012063. [\[CrossRef\]](#)
30. Tresanchez, M.; Pallejà, T.; Palacín, J. Optical Mouse Sensor for Eye Blink Detection and Pupil Tracking: Application in a Low-Cost Eye-Controlled Pointing Device. *J. Sens.* **2019**, *2019*, 3931713. [\[CrossRef\]](#)
31. Aman, Sangal, A. Drowsy Alarm System Based on Face Landmarks Detection Using MediaPipe FaceMesh. In Proceedings of First International Conference on Computational Electronics for Wireless Communications., Haryana, India, 11–12 June 2021 Springer: Berlin/Heidelberg, Germany, 2022; pp. 363–375.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

33. Sifre, L.; Mallat, S. Rigid-motion scattering for texture classification. *arXiv* **2014**. arXiv:1403.1687
34. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**. arXiv:1704.04861
35. Aqib Hashmi, S.; Chowdhury, R.; Mahdy, D. Face Detection in Extreme Conditions: A Machine-learning Approach. *arXiv* **2022**, arXiv:2201.06220
36. Sarang, P. *Artificial Neural Networks with TensorFlow 2: ANN Architecture Machine Learning Projects*; Springer: Berlin/Heidelberg, Germany, 2021.