Titre: Title:	Using a Facial Recognition System Based on Artificial Neural Network Techniques for Assistive Robotic Arm Control.
Auteur: Author:	Elmehdi Elayeb
Date:	2023
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Elayeb, E. (2023). Using a Facial Recognition System Based on Artificial Neural Network Techniques for Assistive Robotic Arm Control. [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/53345/

.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Using Facial Recognition System Based on Artificial Neural Network Techniques for Assistive Robotic Arm Control

ELMEHDI ELAYEB

Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie mécanique

Février 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Using Facial Recognition System Based on Artificial Neural Network Techniques for Assistive Robotic Arm Control

présenté par **Elmehdi ELAYEB**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Aouni LAKIS, président
Sofiane ACHICHE, membre et directeur de recherche
Maxime RAISON, membre et codirecteur de recherche
Abolfazl MOHEBBI, membre

DEDICATION

I would like to express my heartfelt gratitude to my supervisors, Prof. Sofiane Achiche and Maxime Raison, for giving me the invaluable opportunity to work under their guidance.

When I joined the Polytechnique University to pursue my Diplôme d'études supérieures spécialisées (DESS) program in the Department of Mechanical Engineering, Prof. Achiche invited me to join his laboratory. It is largely due to his confidence in me and his unwavering guidance that I successfully completed this stage and moved on to the Master of Engineering program. Prof. Achiche's support throughout all stages of the project and its successful completion has been invaluable, and I am deeply grateful to him.

I am also grateful to Prof. Maxime Raison for accepting to be my co-supervisor. I greatly benefited from his guidance and opinions, which enriched my research.

I would also like to extend my thanks to the students in the laboratory, particularly Gabriel and Yaan, who provided assistance at the outset of this project.

Special appreciation goes to my dear friend Behnam Farsi, who is like a brother to me. Apart from assisting me in my research, he is intelligent, kind, and dependable. I hope to have the opportunity to work with him again in the future.

Last but not least, I would like to express my deepest gratitude to my family and friends, who are the most precious gifts in my life. Their unconditional love, respect, and trust have always motivated me to strive for my goals, chase my dreams, and venture into new horizons. I am truly blessed to have them in my life, and I am thankful for all they have done for me.

RÉSUMÉ

Au cours des dernières décennies, des efforts de recherche ont été déployés pour permettre l'utilisation de robots et de bras robotiques d'assistance pour aider les personnes handicapées physiques à réduire leurs limitations dans les tâches quotidiennes. La plupart de ces travaux de recherche étaient basés sur des méthodes classiques et ont des limites importantes dans leur application. Par exemple, certains ont mis en place des systèmes de contrôle pour aider les personnes handicapées à utiliser des bras robotiques montés sur un fauteuil roulant électrique. Cependant, la recherche part du principe que la personne handicapée peut utiliser ses mains pour contrôler le joystick du fauteuil, ce qui n'est pas toujours possible. Le coût élevé de mise en œuvre et l'imprécision des capteurs sont également des limitations courantes.

L'objectif principal de ce mémoire de maitrise est d'utiliser une technique de réseau de neurones artificiels (ANN) en reconnaissance faciale pour contrôler un bras robotique d'assitance. La reconnaissance faciale est devenue un domaine clé dans l'ANN et le traitement d'images. Dans l'application, les mouvements des yeux et de la bouche sont utilisés pour contrôler le robot afin d'aider l'utilisateur à boire. L'outil d'évaluation pour valider le système mis en place consiste à choisir entre trois gobelets de couleurs différentes et à amener le gobelet sélectionné par la bouche d'un utilisateur. Dans le projet, un environnement de simulation graphique est conçu et une nouvelle méthode de reconnaissance faciale adaptative est mise en œuvre pour supprimer les bruits générés par le clignement ou l'inclinaison du visage de l'utilisateur. De plus, la caractérisation des bibliothèques et la description de l'implantation des logiciels sont présentées, suivies des conclusions et des suggestions pour les travaux futurs. Les résultats en simulation montrent que la vitesse et la précision du système sont des points forts du modèle proposé.

ABSTRACT

Over the past few decades, extensive research has been conducted on the use of robots and assistive robotic arms to alleviate limitations faced by physically disabled individuals in their daily tasks. However, many of these studies have relied on classical methods and are associated with significant limitations. For instance, some previous works have utilized control systems to aid disabled individuals using a robotic arm mounted on a powered wheelchair, assuming that the user can operate the chair's joystick with their hands, which may not always be possible. Other limitations of previous research include high implementation costs and sensor inaccuracies.

The main objective of this master's thesis is to employ an artificial neural network (ANN) approach for facial recognition in order to control a simulated MICO assistive robotic arm by Kinova [5] in a simulation environment. Facial recognition has emerged as a key area in both ANN and image processing. In this application, eye and mouth movements are utilized to control the robot, specifically to assist the user in drinking. The assessment tool to validate the implemented system involves selecting one of three differently colored cups and then bringing the chosen cup to the user's mouth. The project includes the design of a graphical simulation environment, the implementation of an adaptive facial recognition method to mitigate noise from blinking or head tilting, characterization of libraries, description of software implementation, and conclusions with suggestions for future research. The results demonstrate that the system's speed and accuracy are notable strengths of the proposed solution.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN Artificial neural Network

ARM Assistive Robotic Manipulator

APAC Asian Pacific and Australian Continent

BMI Brain–Machine Interface

BSD Berkeley Software Distribution (A family of permissive free software licenses)

CNN Convolutional Neural Network

DESS Diplôme d'études supérieures spécialisées

EAR Eye Aspect Ratio

EEG Electrode Electroencephalography

HMI Human Machine Interface

ID Intention-Driven

IDE Integrated Development Enviroment

MTCNN Multi Task Convolutional Neural Network

RGB Red Green Blue

RoW Rest of the World

SIR Semi-Autonomous Intelligent Robotic

SSD Single shut Multibox Detector

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Context

There are various types of physical disorders that can significantly impact an individual's autonomy and ability to perform daily tasks such as eating and drinking. According to a survey cited in [1], mobility-related disabilities are the most prevalent disorders among Canadians aged 15 years and older, with over 2.6 million people reporting some level of mobility impairment. These conditions not only cause challenges and discomfort for the affected individuals but also result in significant social and medical costs.

Fortunately, there is a growing body of research dedicated to developing various types of robots as tools to enhance the quality of life for individuals with disabilities. In particular, assistive robotics has emerged as a crucial area of research. An assistive robot is defined as a robot that can receive and process data from the user, and then utilize that data to perform actions that benefit the user in their daily life. For instance, robotic assistive arms have shown promise in assisting individuals with upper limb disabilities in performing basic daily tasks, such as drinking, as evidenced by previous studies [2].

Figure 1.1 Assistive arm helping a disabled user to drink [2]

The trade-off between technical features and architectural complexity can pose challenges in the implementation of assistive robotics. While advanced features may be needed for complicated tasks, they can increase costs and operational difficulties for users. For example, touchscreens or joysticks are commonly used for controlling assistive arms, but joysticks may not always provide the precision required for tasks that involve six degrees of freedom. This highlights the need to explore alternative control methods that are both precise and cost-effective.

Users of assistive robotics devices stand to benefit from the growing research field and the expanding market for assistive robotics. The market for assistive robotics is projected to grow significantly in regions such as North America, Europe, Asia Pacific and Australian Continent (APAC), and Rest of the World (RoW), with expected market value to increase from USD 4.1 billion in 2019 to USD 11.2 billion by 2024 [3]. Figure 1.2 below provides a visual representation of the anticipated growth in the assistive robotics market.

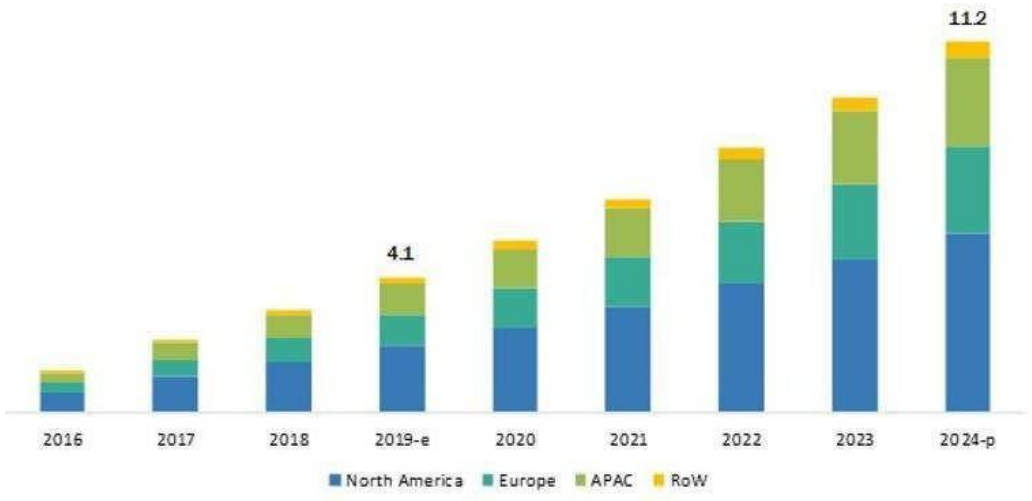

Figure 1.2 Assistive robotics market by region (USD billion) [3]

The significance of utilizing cutting-edge technologies in the development of efficient assistive robots for the future market cannot be overstated. Therefore, the main objective of this master's thesis is to design an accurate control system with minimal implementation costs for assistive arms, specifically tailored to aid users with upper limb disabilities. To achieve this research goal, an extensive review of previous research work, including configurations, results, strengths, and weaknesses, will be conducted. At its core, this thesis aims to develop an assistive robot that

operates swiftly and efficiently, as these criteria are critical for any assistive device. Furthermore, the robot should be user-friendly, with processes that are straightforward and easy for the user to execute. The proposed model in this thesis is not only fast and responsive, but also utilizes the user's mouth and eyes for control, making it highly accessible for individuals with disabilities.

CHAPTER 2 LITERATURE REVIEW

Numerous studies have been conducted to optimize the usability of robotic arms for commercialization as technical aids. Examples of these investigated robotic arms include the MANUS Assistive Robotic Manipulator (ARM) [4], the MICO arm [5], and the JACO arm [6]. Depending on the type and severity of the user's disability, different control methods can be employed, such as a joystick, brain-machine interface (BMI) [7], or 'smart' methods utilizing machine learning and computer vision. In many of the application examples proposed in the reviewed studies, the user has direct control over the full arm path, enabling precise task execution.

For instance, the MANUS robotic arm, as investigated in [8], proposed a control system that utilizes an eye-in-hand and eye-to-hand cooperation scheme, where manual and automatic control of the controller are performed simultaneously. A "one-click approach" is implemented in the multi-camera system to reduce user-operator interaction, where the assistant automatically moves to the desired object on the screen using two cameras and a visual survey design. However, the system's major drawback is its high implementation costs. In [9], another user interface was developed for the MANUS arm, where the user can control the robot using a touch screen and joystick. The researchers primarily focused on developing a Human-Machine Interface (HMI). Figure 2.1 below depicts a photo of the experiment setup that was implemented.

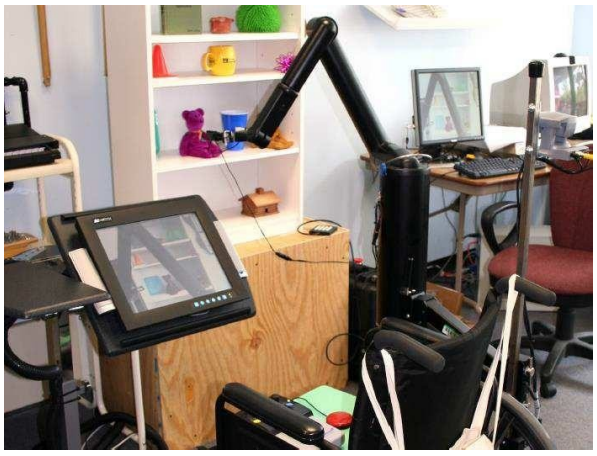

Figure 2.1 Flexible interface for controlling robotic arm experiment setup [9]

The results of the experiment indicate that the controlling system performed as expected, albeit with some observed slowness. However, there are limitations in the design of the Human-Machine Interface (HMI). For instance, the assumption of the shoulder camera view being motionless may not hold true in daily usage, potentially resulting in loss of vision for the user. Moreover, the implemented setup appears to be relatively complex, which presents further challenges. As a result of these limitations and weaknesses, there is a need for the development of new control methods to replace classical ones, such as Brain-Machine Interface (BMI) and facial recognition, which hold promise as potential solutions.

In a study conducted by [10], the focus was on facilitating human-machine interaction by enabling users to easily communicate with robots. Novel approaches were proposed in three different domains: 3D object recognition, prehension, and scene segmentation. The authors tested the models with the JACO arm, and the results showed that the arm was able to retrieve the object within 15 seconds. The paper further proposed novel methods and an ultra-fast decision tree for the recognition and detection of objects, utilizing an effectively designed Artificial Neural Network (ANN) that analyzed an object in a scene in an average of only 0.6 seconds. Furthermore, in the same study, the proposed algorithm was implemented on the JACO robotic assistance arm, which was able to pick up a requested object in 15 seconds while moving at 50 mm/s, indicating a significant improvement in functionality. However, it should be noted that this still required the use of a Kinect camera, which consumes a considerable amount of power [10].

In another study by [11], a robot controlled via Brain-Machine Interface (BMI) was implemented, incorporating a novel technique to estimate the exact location of the user's mouth to assist with drinking. To achieve this objective, a sensor for mouth and cup detection, as well as recording systems for identifying user commands, were employed. Several trials were conducted to evaluate the robot's ability to assist a disabled person in drinking, and all 13 runs were successful. However, the average running time per experiment was 3 minutes, which could be considered a limitation. Additionally, the study noted that the user's body was not fully detected, potentially resulting in unreliable collision avoidance capability and system slowdowns. Another drawback mentioned was the high cost of the RGB-D Kinect sensors used in the study [11].

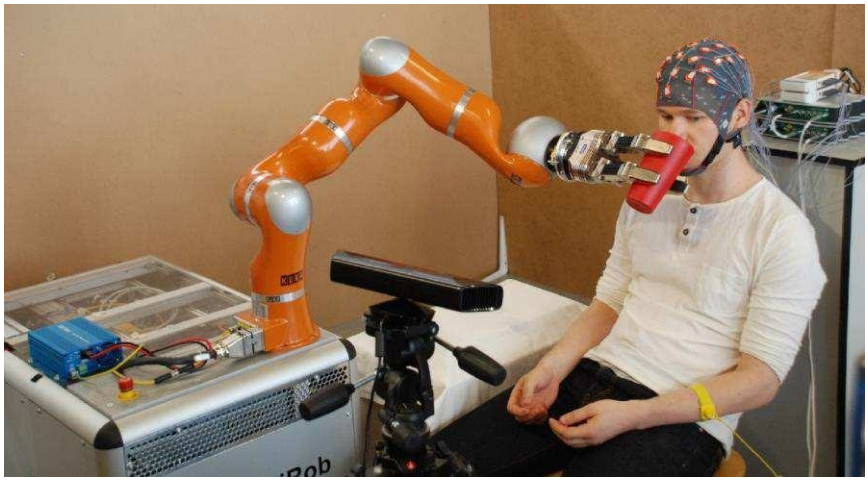

Figure 2.2 BMI-controlled robot helping a user to drink [11]

In [12,13], novel BMI-based control systems are proposed. In [12], reinforcement learning is utilized, and the controlling commands are obtained through dry-electrode electroencephalography (EEG) during imaginary movements. The BMI setup is tested on five individuals aged 23 to 30, and the accuracy of the results ranges from 60.6% to 74.4%, depending on the individual's ability to absorb and correct mistakes. In [13], an adaptive control system is used to enable a BMI to perform goal-directed navigation of a simulated wheelchair. The researchers found that adaptive control significantly improves the performance of the BMI system, with a 50-second reduction in running time compared to previous robots in the experiment. However, it should be noted that the implemented system is complex and expensive [13].

In [14], another BMI-based system for reaching and grasping a foam ball is proposed. Although the system has a fast-running time of about 7 seconds, the grasping accuracy is reported to be only ~62%. This suggests that the performance of this approach is still limited and may not be suitable for commercial applications due to low execution accuracy. The performance metrics of this algorithm need to be improved for real-world use cases.

Another approach involves facial recognition-based algorithms. Facial recognition aims to quickly and accurately identify human faces in images or videos, emulating human perception. There are various feature-based methods for face recognition [15][16].

In the following sections, research on vision-based subsystems using different face recognition methods is reviewed. In [17], an intention-driven intelligent robotic (ID-SIR) system is

developed, which consists of a BMI subsystem, visual detection, and arm. The system combines feature extraction and matching processes by training a Convolutional Neural Network (CNN) to distinguish between four pre-defined objects. It is implemented on the Kinova JACO assistive arm with 6 degrees of freedom. The test reports show a running time of 84 seconds. In comparison with BMI-based systems [11,14], the ID-SIR system demonstrates significantly higher accuracy, with grasping, delivering, and drink task accuracy reported at ~97.5%. However, it should be noted that the use of two Time-of-flight Kinect cameras makes this setup expensive. Figure 2.3 below shows a photo of the setup and the defined tasks for the ID-SIR system.

Figure 2.3 ID-SIR system setup [17]

The authors in [18] present an eye tracking-based robotics control study that aims to combine eye tracking and computer vision to automate the approach of a robot to its target point by obtaining its 3D location. By utilizing image processing methods, the accuracy of the task was improved by 92% compared to using eye tracking alone [18].

In [19], an affordable prototype of a face-recognizing assistive arm designed to assist users in drinking is developed. The setup employs two cameras, one for capturing objects and the other for capturing the user's face. A face recognition algorithm is implemented in this study, enabling

the system to detect the user's face and visually control the arm. Although the preliminary tests confirm the algorithm's performance, the running time is not reported [19].

In [20], a novel and promising AdaBoost face detection algorithm is proposed. This algorithm employs resizing and skin filter technology to detect faces and has the potential to be used in commercially available assistive arms. The reported results demonstrate that the error rate decreases to almost zero after a few iterations, highlighting the benefits of using machine learning-based methods [20]. Another similar eye-tracking method is proposed in [21]. This technique employs image processing to process gaze signals for controlling a human-robot interaction. The results show that the system is not only fast (with reduced reading time from 4,840 ms to 3,000 ms), but it can also accurately distinguish gaze clusters with 83% accuracy. However, one weakness of the system is the requirement of using a joystick as well. Figure 2.4 depicts the setup described above [21].

Figure 2.4 Eye gaze detection setup [21]

In [22], a novel artificial stereo vision and eye-tracking system called PoGARA is proposed for controlling an assistive arm, utilizing the JACO arm in the setup. The results demonstrate impressive processing speed and high accuracy of the system. Specifically, the average time for reaching and grasping an object was recorded as 13.8 seconds with obstacles and 17.3 seconds without obstacles. The accuracy of the PoGARA system was reported to be higher than 90%.

2.1.1 Review of Convolution Neural Network-Based Algorithms

The multi-task deep convolutional neural network (MTCNN) introduced by Zhang K et al [23] is considered one of the most advanced CNN structures for facial recognition tasks. It employs a three-stage process to recognize the face and detect five key facial landmarks. In the first stage, a rapid CNN called P-net generates candidate regions of the face. The second stage involves a refinement network (R-net) that fine-tunes the facial regions. Finally, the output network (O-net) in the third stage generates the bounding box and precise positions of the facial landmarks [23]. MTCNN is one of the commonly used methods in facial recognition software implementation, and further details on these methods can be found in Chapter 3.

CHAPTER 3 OBJECTIVE OF THE WORK

The literature review demonstrates that significant research efforts have been dedicated to developing efficient solutions for controlling assistive arms. However, there are limitations in the implemented systems, including inadequate software for accurate and responsive model detection, as well as processing time challenges in computer vision projects. Therefore, the objective of this master's thesis is to propose a simple approach that overcomes these limitations by utilizing face features such as eyes and lips for performance-driven control of the assistive robotic arm.

The present study aims to enhance the capabilities of the assistive robotic arm through the development of a software-based vision system that utilizes eye and mouth movements for control. The vision system is designed to accurately detect the user's face from a noisy environment, using OpenCV, an open-source machine vision library written in C++. The proposed approach combines digital image processing and deep learning techniques within the field of computer vision [24].

CHAPTER 4 EXPERIMENTAL IMPLEMENTATION OF NOVEL FACIAL RECOGNITION SYSTEM TO CONTROL ASSISTIVE ROBOTIC ARM

The thesis aims to develop a facial recognition system as a controlling tool for an assistive robotic arm, specifically to bring a cup to a human subject's mouth autonomously. The system will implement the developed algorithm on a simulated assistive robotic arm, using facial recognition to detect the closing of the eyes and opening of the mouth as cues for choosing the cup for the human subject.

Image processing has become a crucial tool in engineering, with numerous applications, including facial recognition. Many machine learning and deep learning models have been proposed for facial recognition. In this chapter, various learning algorithms that detect the status of the user's eyes and mouth (open or closed) are discussed. Face detection approaches are first reviewed, as accurately detecting the face is essential for eye and mouth detection. Subsequently, different learning models are discussed to identify the most suitable model for eye and mouth detection. Despite existing models in the literature, the thesis proposes a new deep learning model and compares its accuracy with other models. The results demonstrate that the proposed model surpasses some limitations of existing models, exhibiting superior performance in eye and mouth detection.

This chapter is organized as follows:

1. Section one studies the structure of deep neural networks and convolutional neural network models.
2. Section two explains the implemented model for face detection.
3. Section three talks about predicting the landmarks of important regions in the detected face.
4. In section four, two existing deep learning models are discussed, and a new model is proposed to detect whether eyes and mouth are open or closed.
5. The results of these investigations and of the chapter as a whole are provided in section five.

4.1 Face-Detection Model

The Single Shot Multibox Detector (SSD) with ResNet-10 architecture is a ready-made deep neural network that can be used for detecting the face of the user in the thesis project [25]. SSD is an object detection algorithm developed by Google, which builds on the VGG-16 model introduced by Simonyan et al. in [26]. VGG-16 is a CNN model that achieves high accuracy of up to 92.7% in the ImageNet database, which contains over 15 million labeled high-resolution images [27].

In [28], SSD is presented as an architecture that outperforms state-of-the-art CNNs, including Faster R-CNN, which is based on earlier work by Szegedy et al. [29] on MultiBox, a quick class-agnostic bounding box coordinate proposal. SSD is a single deep neural network approach for object detection. The method involves discretizing the output space of bounding boxes into a set of default boxes with different aspect ratios. These fixed-size bounding boxes are then passed through feature maps in a feed-forward convolutional network [30]. VGG-16 is used to create the first layers of SSD, and an auxiliary structure is added to the network from conv6 to extract features at multiple scales and reduce the input size to each layer [31].

During the prediction process, a confidence score is generated for each detected object, and the network makes adjustments if the object belongs to the object classifiers. The confidence score reflects how close the prediction is to the actual object, particularly for the opening and closing of the eyes and mouth in this case. The loss function for SSD is the sum of confidence loss and localization loss. The localization loss measures the deviation from the ground truth box, while the confidence loss indicates the level of confidence for the abjectness of the computed bounding box [32]. Figure 4.1 provides an illustration of the architecture of SSD [25].

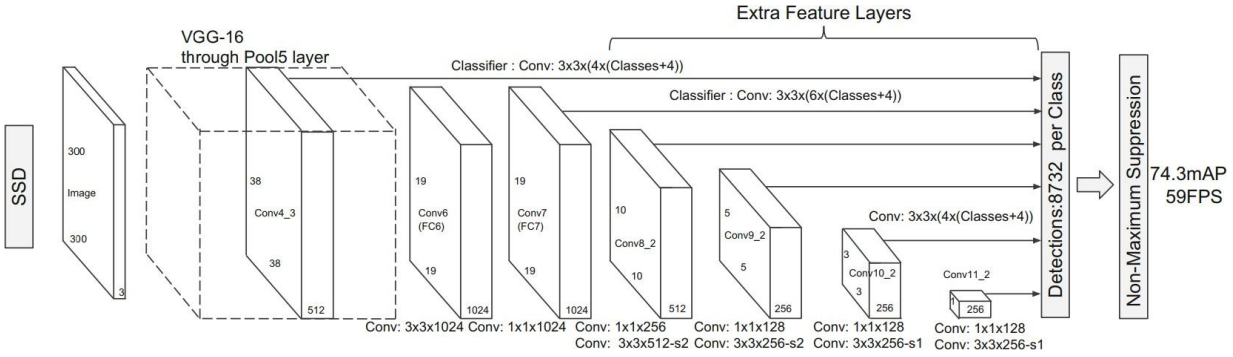

Figure 4.1 The architecture of SSD [25]

4.2 Landmark Prediction.

The structure of the proposed algorithm of face recognition that has been used in this master's thesis as shown in figure 4.2.

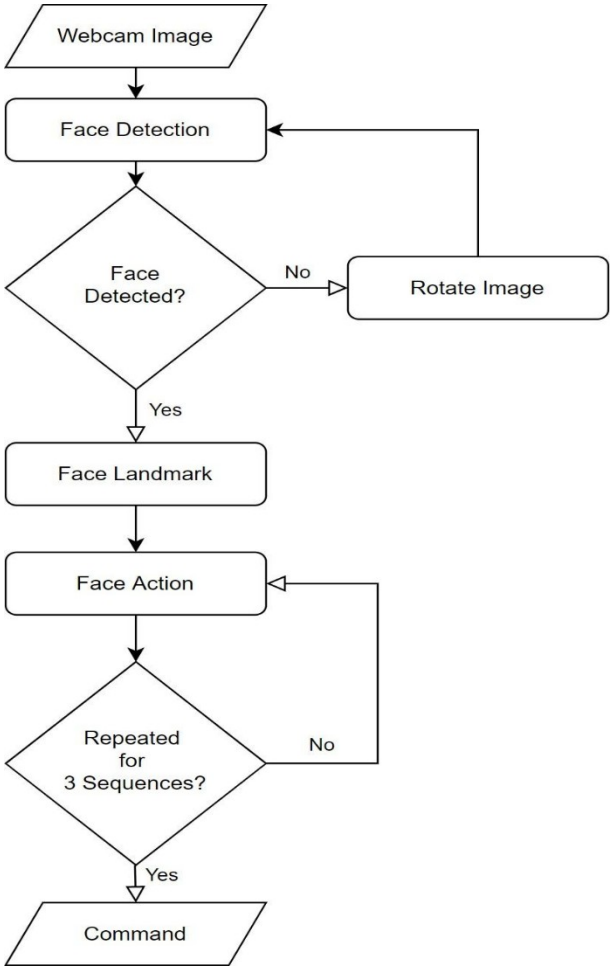

Figure 4.2 The structure of proposed algorithm

The previous section explained how an SSD Model is able to detect the face of the user. To detect the face regardless of its angle, relative angles of ±30, ±60, and ±90 degrees are first checked to find a face. Next, a rectangle is drawn over the detected face. The detected picture is then rotated to be properly vertical, using affine transform matrices. The same transformation is employed for rotating, translation, and scaling. This method uses two points as X and T, in which X is the coordinates of a point in the input picture and T is the desired coordinates of X in the output picture. Having the 2×3 transformation matrix M and input X, T can be calculated using equations (4-1 to 4-3). Every other coordinate can also be mapped to the desired coordinates using the same equations [33].

$$X = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, T = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \qquad (4-1)$$

$$M = \begin{bmatrix} a_{00} & a_{01} & b_{00} \\ a_{10} & a_{11} & b_{10} \end{bmatrix} \qquad (4-2)$$

$$T = M. \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \qquad (4-3)$$

$$M = T. \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}^{-1} \qquad (4-4)$$

If the input and output are known, M can be calculated using equations (4-2 to 4-4). Since 6 constants define the general transformation M, M can only be calculated with 3 pairs of input and output (6 equations with 6 unknown parameters). In this algorithm, the three pairs are the three corners of the picture [33].

The algorithm for detecting the eyes using the Haar Cascade Model and the Dlib 68 landmark model [34] involves the following steps:

1. After the face is detected using the SSD model, the Haar Cascade Model is used as classifiers to detect the eyes. This involves drawing rectangles around the eyes based on the trained classifiers.
2. At the same time, the Dlib library is used to split the detected face into 68 landmark points. These landmark points are used to identify distinct regions of the face, such as the face frame, eyebrows, eyes, nose, and mouth. Landmark points 17, 5, 6, 9, and 20 are specifically allocated to these regions.

3. Each group of designated landmark points is colored distinctly to make them separable for further analysis and processing.
4. The algorithm also calculates the angle of the face from the rotated face. This is done by drawing a line between the left and right eye landmarks and measuring the angle of this line with respect to the horizontal axis. Any deviation from the horizontal axis indicates the angle error of the face.
5. Since the face detection algorithm's rotation resolutions are limited to 30°, the calculated angle in this section is added to the previously computed angle from the face detection step to obtain a more accurate face angle.

Figure 4.3, as mentioned in the text, shows a photographic image of eye landmarks for visual reference [35].

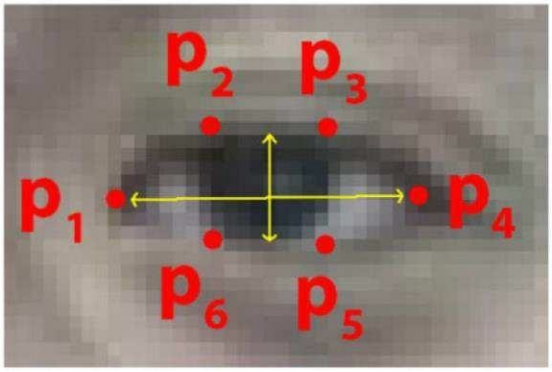

Figure 4.3 Pixelated image of eye landmarks [35]

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \qquad (4-5)$$

The proposed algorithm in this master's thesis for eye and mouth detection involves the following steps:

1. First, the Haar Cascade Model for the eyes is used to detect rectangles around the open eyes, as shown in figure 3.3. This step helps in identifying the eyes and their status (open or closed).
2. Next, the Eye Aspect Ratio (EAR) is calculated using the Dlib Model. EAR is the average coordinates of the six landmarks of each eye, and it is calculated using equation (4-5) in

the Dlib Model [36]. The EAR is used to determine if the eyes are open or closed. If the average coordinates fall inside one of the rectangles detected in the previous step, it indicates that the eyes are open, otherwise, they are closed.

3. The mouth is evaluated using the aspect ratio of the outer-mouth (lip) landmark, as it provides sufficient accuracy for the application.
4. A threshold value is used to determine the status of the eyes and mouth. The value of the threshold is obtained through trial and error to achieve the desired accuracy.
5. To eliminate noise or interference in the process, a technique called debouncing is implemented. For example, if the algorithm detects an open mouth, it will not immediately change the mouth's status. The algorithm will only operate once if one of the commands for opened-mouth, closed-left-eye, and closed-right-eye were repeated for more than three instances. This helps in avoiding mistakes and ensuring accuracy in the process.
6. The camera continuously monitors the user's face and detects the movements of the eyes and mouth based on the explained learning approaches. Once a blinking or mouth opening occurs, the deep learning model detects the movement and sends the defined commands to the robot arm for further action.

The mapping between face movements and robot movements is achieved through this algorithm, which continuously monitors the user's face and detects eye and mouth movements to trigger corresponding commands for controlling the robot arm, as explained in further sections [37].

4.3 Data Set

The evaluation and comparison of different models for image processing in the developed algorithms are crucial to determine their performance. To conduct this evaluation, a data set containing 2,900 face images is used. These images are used to train and test the models for classification into four different categories: 1) Open eyes, 2) Closed eyes, 3) Yawn, and 4) No-yawn.

The data set is divided into two sets: a training set and a testing (validation) set. The training set consists of 85% of the data, which is used to train the deep learning models. The models are trained on images of eyes and mouths, both open and closed, using this training data set.

Once the training is completed, the testing data set, which comprises the remaining 15% of the data, is used to evaluate the performance of the trained models. The models are tested on this data set to analyze their accuracy, precision, recall, F1-score, and other relevant performance metrics.

By using this approach of dividing the data set into training and testing sets, the performance of the models can be assessed on unseen data, helping to determine their effectiveness in accurately classifying the images into the four categories mentioned above. This evaluation process aids in selecting the most suitable model for the image processing section of the algorithm based on its performance on the testing data set and allows for any necessary adjustments or fine-tuning to be made to improve the model's performance.

Table 4.1 Data set of the eye with 6 points and the mouth with 20 points

Data	**Eyes**		**Mouth**	
Situation	**Open**	**Close**	**Open**	**Close**
Image #1	True	False	False	True

The above dataset shows how Model is supposed to make prediction for eyes and mouth against the actual data. It shows how models predict the output to actual classification of eyes and mouth.

4.4 Deep Learning Models

The VGG19 model is chosen as it is a widely used convolutional neural network (CNN) architecture developed by the Visual Geometry Group (VGG) at the University of Oxford, known for its excellent performance in image classification tasks. VGG19 is a deeper version of the original VGG16 model, with 19 layers, including 16 convolutional layers and 3 fully connected layers [38].

The VGG19 model is specifically suitable for detecting and tracking eyes and mouth in face images, as it is designed for image classification tasks and has been proven effective in detecting complex features from images. The convolutional layers in VGG19 are responsible for extracting features from the input images. These convolutional layers use small filters with a small receptive field, which allows them to capture local patterns and details in the images. The use of multiple convolutional layers with small filters helps in capturing hierarchical features of increasing complexity, which is beneficial in identifying eyes and mouth in face images.

To further improve the feature extraction process, VGG19 incorporates max pooling layers, which help in down-sampling the feature maps and reducing the spatial dimensions while retaining the most important features. Max pooling is a non-linear operation that selects the maximum value from a group of values, which helps in retaining the dominant features while discarding less significant ones [39].

After the feature extraction process, the extracted features are passed through three fully connected layers in VGG19. These fully connected layers are responsible for converting the features into a format that can be used for classification. Each fully connected layer has its own set of weights, which are learned during the training process, and is followed by an activation function [40]. In VGG19, the Softmax activation function is used in the last fully connected layer, which is suitable for multi-class classification problems like the one in this thesis. Softmax function normalizes the output probabilities based on the normal distribution, providing the probability distribution for each class, which helps in determining the class with the highest probability [41].

Overall, the VGG19 model is a powerful deep learning architecture which is shown in figure 4.4 that is capable of effectively extracting features from images and performing image classification tasks. Its deep architecture, use of convolutional and max pooling layers, and the Softmax activation function make it well-suited for the task of detecting and tracking eyes and mouth in face images in this master's thesis [42].

- Conv3x3 (network of size 64)
- Conv3x3 (network of size 64)
- Max Pool layer
- Conv3x3 (network of size 128)
- Conv3x3 (network of size 128)
- Max Pool layer
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 256)
- Max Pooling layer
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 256)
- Conv3x3 (network of size 512)
- Max Pooling layer
- Conv3x3 (network of size 512)
- Conv3x3 (network of size 512)
- Conv3x3 (network of size 512)
- Conv3x3 (network of size 512)

- Pooling layer
- Fully connected layer (4096)
- Fully connected layer (4096)
- Fully connected layer (1000)
- Soft Max activation function

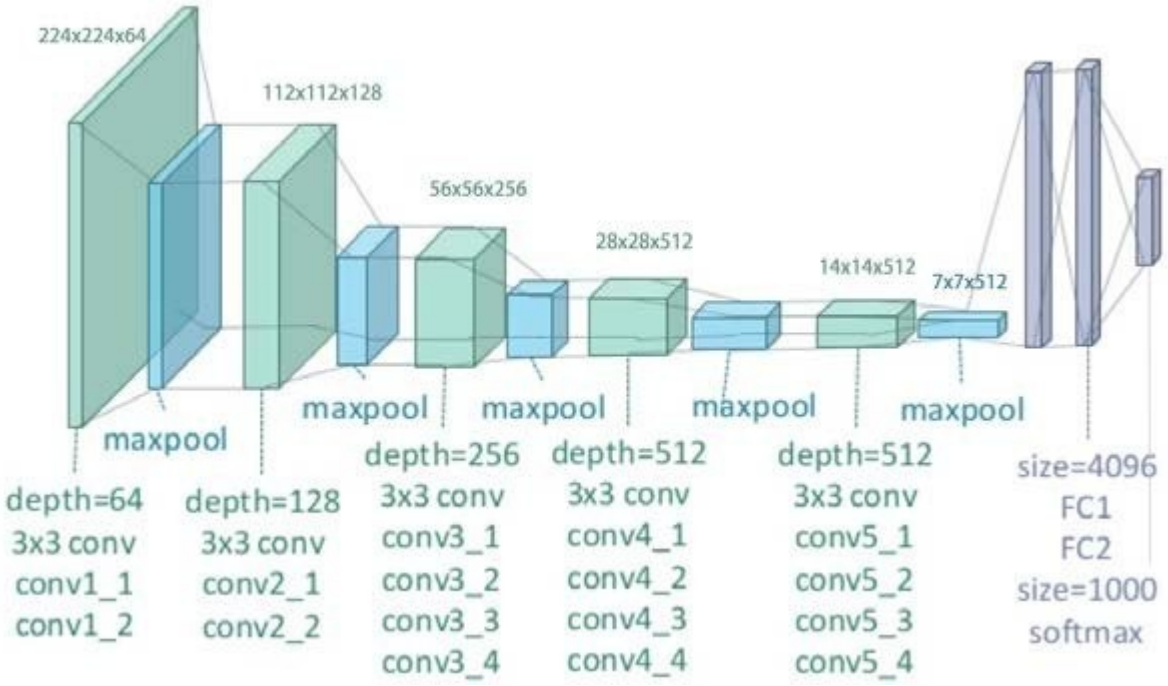

Figure 4.4 llustration-of-the-network-architecture-of-VGG-19-model [42]

4.5 Implementation

The VGG19 model is a powerful pre-trained deep learning model that has been trained using the ImageNet dataset, making it one of the most widely used and effective models for various classification problems. It consists of a collection of convolutional and dense layers, where dense layers are also known as fully connected layers, as each neuron in these layers receives input from all the neurons of the previous layer. These dense layers help in understanding hidden patterns in the data and provide the expected output after passing through an activation layer [43].

The input image size required by VGG19 is fixed at (224x224) RGB images [42]. The input image is converted into a NumPy array with specified dimensions of samples, rows, columns, and channels. This conversion allows the image to be processed by the VGG19 model, which scales the pixel data appropriately for classification.

In the VGG19 model, the convolutional and pooling layers are considered as the feature extraction part, while the dense layers are considered as part of the classification process. The pooling layers help in down-sampling the feature maps, reducing the spatial dimensions, and retaining the most important features. The convolutional layers, on the other hand, are responsible for extracting features from the input image, capturing local patterns and details through small filters with small receptive fields [44].

Additionally, the VGG19 model provides an option to include or exclude the dense layers during model training, through a parameter called "include_top". When set to "False", the dense layers are removed, allowing for customization of the model architecture based on specific requirements.

Overall, the VGG19 model is a well-optimized and widely used deep learning architecture, with pre-trained weights from the ImageNet dataset, making it a powerful choice for various classification tasks. Its combination of convolutional and dense layers, along with the option to customize the architecture, makes it suitable for a wide range of image processing applications.

4.6 Random Forest Algorithm (Eye Tracking)

The random forest algorithm is an ensemble learning technique that leverages the power of multiple decision trees to create a robust and accurate classifier for classification tasks. It is particularly well-suited for tasks such as eye movement detection, which are not only important but also challenging to achieve accurately [45].

One of the key advantages of the random forest algorithm is its ability to parallelize the computation of building trees. In a random forest, multiple decision trees are constructed in parallel, with each tree trained on a random subset of the training data. This parallelization allows for efficient processing of large datasets, as each tree can independently compute branches, sub-branches, and leaves based on information gain or entropy criteria.

The decision tree construction process in random forest involves recursively splitting the data into branches and sub-branches based on the feature values that result in the best information gain or entropy. This process continues until a stopping criterion, such as reaching a certain depth or

having a minimum number of samples in a node, is met. The leaves of the decision trees contain the final classification labels [46].

Once all the trees are constructed, the random forest algorithm combines the predictions of individual trees to obtain the final classification result. This is typically done by averaging the predictions of all the trees in the forest. This aggregation of predictions helps to reduce the risk of overfitting and improve the overall accuracy and robustness of the model.

One of the notable advantages of the random forest algorithm is its ability to handle high-dimensional data with multiple features, which makes it suitable for image processing tasks such as eye movement detection. Additionally, random forests are less prone to overfitting compared to individual decision trees, as the ensemble of trees helps to reduce bias and variance in the model.

The computation time required for building a random forest is generally similar to that of building a single decision tree, as the trees are constructed in parallel. This makes random forest a computationally efficient option for large datasets.

In summary, the random forest algorithm is a powerful and effective technique for image processing tasks, such as eye movement detection, due to its ability to parallelize tree construction, handle high-dimensional data, and reduce overfitting. Its ensemble nature and parallel processing make it a popular choice for achieving accurate and robust classification results [45] [46].

The features we chose to extract from the eyes are as follows:

1. Blinks
2. Pupil dilation
3. Movement of eyes

The eye movement detection process involves continuously tracking the movement of the eyes, including eye blinking and pupil dilation, using techniques such as distance formulas to calculate the distance of pupil movement. These eye features, obtained from the distance formulas, are then used as input to build a classifier using the scikit-learn library.

Before building the random forest model, a decision tree is constructed to determine the best approach for the specific task. Random forests consist of parallel-built decision trees, which make them highly efficient and effective for achieving better model performance compared to individual

decision trees. Random forests also offer various hyperparameters that can be tuned to improve the accuracy of the model and training speed.

To ensure optimal training of the model, different training and test data are used in cross-validation. The K-fold cross-validation technique is commonly used, where the data is divided into k folds. In this case, a k value of 4 is used, which means that the data is divided into four sets for training and testing. The average accuracy of the model is calculated for each individual set to identify the strongest models among those tested.

This approach allows for robust model evaluation and selection, as it accounts for variations in the training and test data and helps to prevent overfitting. By leveraging the power of random forests, along with appropriate hyperparameter tuning and cross-validation, the accuracy and efficiency of the eye movement detection model can be improved, leading to more accurate and reliable results in the study.

4.7 Convolutional Neural Network (Mouth Tracking)

A convolutional neural network (CNN) is a type of neural network that is inspired by the human brain's nervous system and is commonly used for tasks such as image recognition, computer vision, and medical image classification. It consists of sequential, dense, and pooling layers that together form a neural architecture designed to identify patterns in input images [47].

CNNs are capable of dealing with images of any dimension, making them well-suited for computer vision and facial recognition applications [48]. They typically have four main parts: the input layer, pooling layer, hidden layer, and activation layer. CNNs use filters to learn from the input data provided, and the activation function determines the output of the network based on the hidden layers' output. The spatial size of the input is often reduced to mitigate overfitting, and a fully connected layer transforms the result from the feature learning part to the output [49].

For the development of a mouth-tracking model, a 2-dimensional CNN was trained on pre-processed mouth samples, and their different parameters were compared. The training set consisted of pre-processed and transformed image data, which was used to train the model.

During the training process, the input layer reads the patterns present in the images, and the convolutional layer learns the parameters. The term "parameters" refers to the learnable elements of the network. The total number of parameters in the CNN network can be calculated using the

formula ((kernel size x stride + 1) x filters), with 1 being the bias added to each filter. In this case, the total number of parameters is 6,048 [50].

The CNN layer in this model has 48 filters, and in the second layer, the total number of parameters is 1,536,256. As a supervised learning algorithm, the CNN is capable of classifying the images into the desired 'open' and 'closed' classes, forming a binary classification. The total number of parameters involved in this training process using this architecture is 50,870,451 [51].

Batch normalization is implemented in the network architecture, which is an efficient method to normalize the representation of data internally, improving the speed of computation, overall performance, and training of the neural network architecture [25]. In this work, the main benefit of using batch normalization is to achieve stochastic optimization during the training process, as it regularizes the model and normalizes data in smaller chunks. This technique also reduces the internal covariance shift, which can occur due to changes in weights at each layer, and allows for a higher learning rate, speeding up the learning process.

Throughout the training process, five main points were observed in the network, from the beginning to the end.

1. Normalization was performed for every batch before providing input data into each neural network.
2. A batch size of 16 was given as an input image.
3. Input data were transformed from the NumPy array of values of float16 between 0 and 255.
4. A test between the models containing normalized data and not normalized data is carried out.
5. Normalization aims at providing a 5% improvement in the final accuracy of the model.

Dropout is a one of the major regularization techniques which aims at reducing over-fitting and improve the function of neural networks. The proposed model architecture, with a dropout rate of 25%, had improved test accuracy of more than 5% compared to when the dropout was not present. In the same way, using an even higher dropout rate of 45%, the validation accuracy increased by around 5%. The validation accuracy was higher than the test accuracy, which proves that over-fitting was successfully prevented. Therefore, the overall model accuracy was improved by adding dropout as one of the critical techniques.

4.8 Implementation Tools

The following software, models, and libraries are used in the implementation process of the system:

- Kinova MICO robotic arm model as the assistive robotic arm [5].
- CoppeliaSim software and PyRep library to simulate the process [52][53].

4.8.1 MICO Assistive Arm

The MICO robotic arm is a cutting-edge technology that is powered by a 24-volt battery or a standard electrical plug. It is designed to be highly versatile and adaptable for various applications, making it suitable for both daily activities and advanced robotics research. One of its notable features is its four degrees of freedom (DOF) model, which includes a gripper, as shown in Figure 3.4. This model is chosen for its availability and compatibility with the CoppeliaSim simulation environment, which allows for accurate simulation and testing of the robotic arm's capabilities [52].

The MICO arm is equipped with two under-actuated fingers, which enable it to safely handle everyday objects with precision and control. This makes it ideal for tasks that require delicate manipulation, such as grasping and manipulating objects in a human-like manner. The arm's gripper is capable of gripping and releasing objects with high dexterity and accuracy, making it suitable for a wide range of applications, from industrial tasks to assistive devices for individuals with mobility issues [5].

One of the notable advantages of the MICO robotic arm is its modular design, which allows for easy customization and expansion. The arm can be easily mounted on different platforms, such as wheelchairs, to assist individuals with mobility limitations in performing various tasks independently. This makes it a valuable tool for enhancing the quality of life for individuals with disabilities and providing them with greater independence and autonomy in their daily activities.

Furthermore, the MICO arm is known for its efficiency, with a payload-to-weight ratio that is optimized for a variety of tasks. Its compact and lightweight design allows for easy integration into different environments and setups, making it a versatile solution for a wide range of applications in robotics research, industrial automation, and assistive technology.

In addition to its physical capabilities, the MICO arm also provides technical specifications in the CoppeliaSim simulation environment, allowing for accurate modeling and simulation of its performance. This makes it a valuable tool for researchers and developers to test and optimize the arm's performance in a virtual environment before implementing it in real-world applications.

Overall, the MICO robotic arm shown in figure 4.5 is a state-of-the-art technology that offers advanced capabilities, modularity, efficiency, and versatility, making it an ideal choice for a wide range of applications in robotics research, industrial automation, and assistive technology. Its high precision, adaptability, and ease of integration make it a valuable asset for enhancing human-robot interactions and improving the quality of life for individuals with mobility limitations.

Figure 4.5 MICO 4-DOF robotic arm [5]

4.8.2 Robot Simulation Environment

CoppeliaSim [52] is a comprehensive robot simulation environment that comes with an Integrated Development Environment (IDE). It is built on a distributed control structure, where each object and model can be controlled independently using various technologies. Controllers can be written in different programming languages, providing flexibility and versatility in implementing robot behaviors.

The distributed control structure of CoppeliaSim makes it highly suitable for multi-robot applications, allowing for the simulation and testing of complex scenarios involving multiple robots interacting with each other and with their environment. This makes it a powerful tool for researchers and developers working on multi-robot systems.

In addition to its versatility, CoppeliaSim is known for its efficiency and performance. It provides a powerful platform for implementing and developing new methods, particularly in the fields of robotics, remote control, and industrial automation systems [52]. The simulation environment allows for fast and accurate modeling of robot behaviors and interactions with the environment, making it a valuable tool for testing and optimizing robot algorithms before implementing them in real-world applications.

To further enhance its capabilities, CoppeliaSim can be linked to vision algorithms using the PyRep library, developed by Stephen James [54]. This allows for seamless integration of vision-based algorithms with the simulation environment, enabling realistic simulations of vision-guided robotic tasks. The Python code can be executed simultaneously with the simulator environment at a high speed, allowing for efficient testing and validation of vision-based algorithms.

It's worth noting that PyRep has some limitations, such as being compatible only with Linux environment and not with Windows [53]. However, despite this limitation, PyRep remains a powerful tool for incorporating vision-based algorithms into the CoppeliaSim simulation environment, enabling researchers and developers to conduct sophisticated simulations and testing of vision-guided robotic systems.

4.8.2.1 Graphical User Interface

A Graphical User Interface (GUI) is implemented to illustrate the developed face recognition algorithm's output. By winking with the left or right eye at the camera, an indicator is navigated to the left or the right between four cups, and then the desired cup can be chosen by opening the mouth.

After running the codes, three different windows are provided. One of them distinguishes the user's face from the surrounding environment and draws a rectangle around the user's face (Figure 4.6). The second window (Figure 4.7) magnifies the selected rectangle and shows the face's 68 landmarks with distinct colors. The third window (Figure 4.8) illustrates four cups. In the third window, the first cup is gray. Using this cup aims to visualize the process of when a user regrets his/her choice and decides to return the cup to its initial position during the cup's movement to his/her mouth. Appendix A shows the detailed code of the GUI in this section.

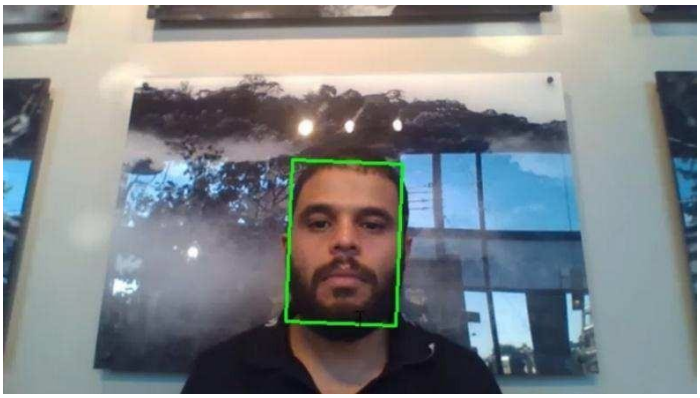

Figure 4.6 GUI's first window detecting the user's face

Figure 4.7 GUI's second window with 68 landmarks

Figure 4.8 GUI's third window simulation four cups

Figures 4.9 and 4.10 illustrate the process of choosing a cup. The facial gesture of the subject is shown on the left, while on the right is displayed the four cups' simulation alongside the currently chosen cup and desired cup.

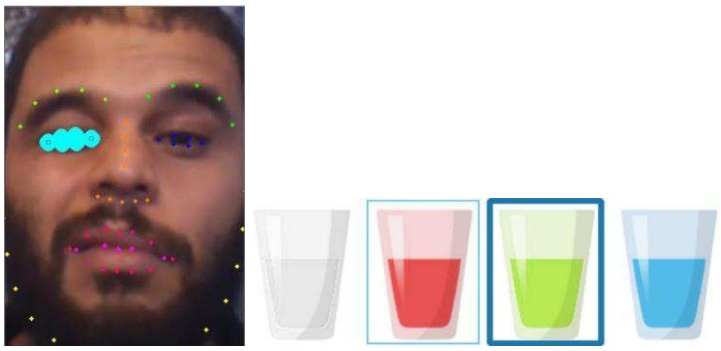

Figure 4.9 Process of moving to the left cup

Figure 4.10 Process of choosing a desired cup

Figure 4.11 illustrates the ability of the algorithm to distinguish a wink from a blink. The picture was captured when the right eye was closed quickly as a natural human response, while the user intentionally closed his left eye. Figure 4.12 shows how changing the face angle does not affect either face detection or landmarks.

Figure 4.11 Distinguishing a wink from a blink

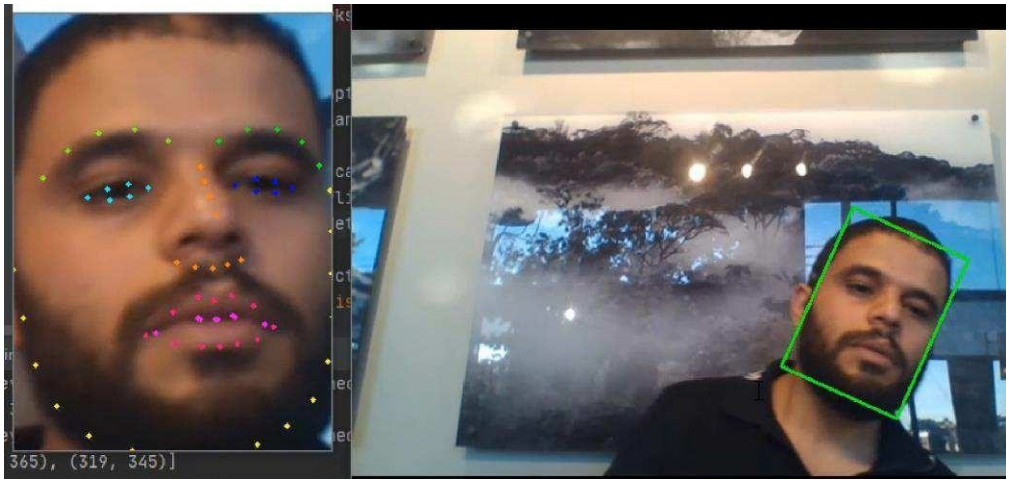

Figure 4.12 Changing of the face angle

4.8.2.2 3D Graphic Simulation

In this section, the proposed algorithm is implemented on a simulated version of the MICO robot in the CoppeliaSim environment. The Python code is connected to CoppeliaSim using the PyRep platform. The user is able to choose a cup by utilizing their eye and mouth movements, and the robot reaches for the desired cup and brings it close to the user's mouth. The simulation environment consists of three colored cups placed on a table (Figure 4.13). The MICO robot model is available as a ready-made model in the non-mobile robots' section of CoppeliaSim, along with other robot models. The floor, lights, and camera can be modified, as well as the six links and joints of the robot. By clicking on the MICO object in the scene hierarchy tab, a Python script can

be opened to plan the robot's movements. A gripper model is also provided in CoppeliaSim for the MICO robot, which allows it to grasp and move cups. Figure 4.14 shows the 4-DOF model of the MICO robot with the gripper.

Figure 4.13 Three cups on a table

Figure 4.14 4-DOF MICO robot with gripper

Figure 4.15 shows the gripper in the CoppeliaSim environment. A script entitled “MicoHand” can be modified for the robot to operate with the desired grasping characteristics. Based on the CoppeliaSim software instructions on efficient grasping, there are two methods of grasping an object. One is to try to model the performance as realistically as possible. This requires delicate adjustment of parameters such as motor forces/torques/velocities, friction coefficients, object masses, and inertias. The other method is to attach the cup to the gripper with a connector whenever the robotic hand reaches it. The implementation is more straightforward this way, and

the results are more stable. However, even though it is simpler, it is not always realistic, as this approach must be tested with real-world data and arm, which is a future work. Despite these limitations, the second method is used in this thesis.

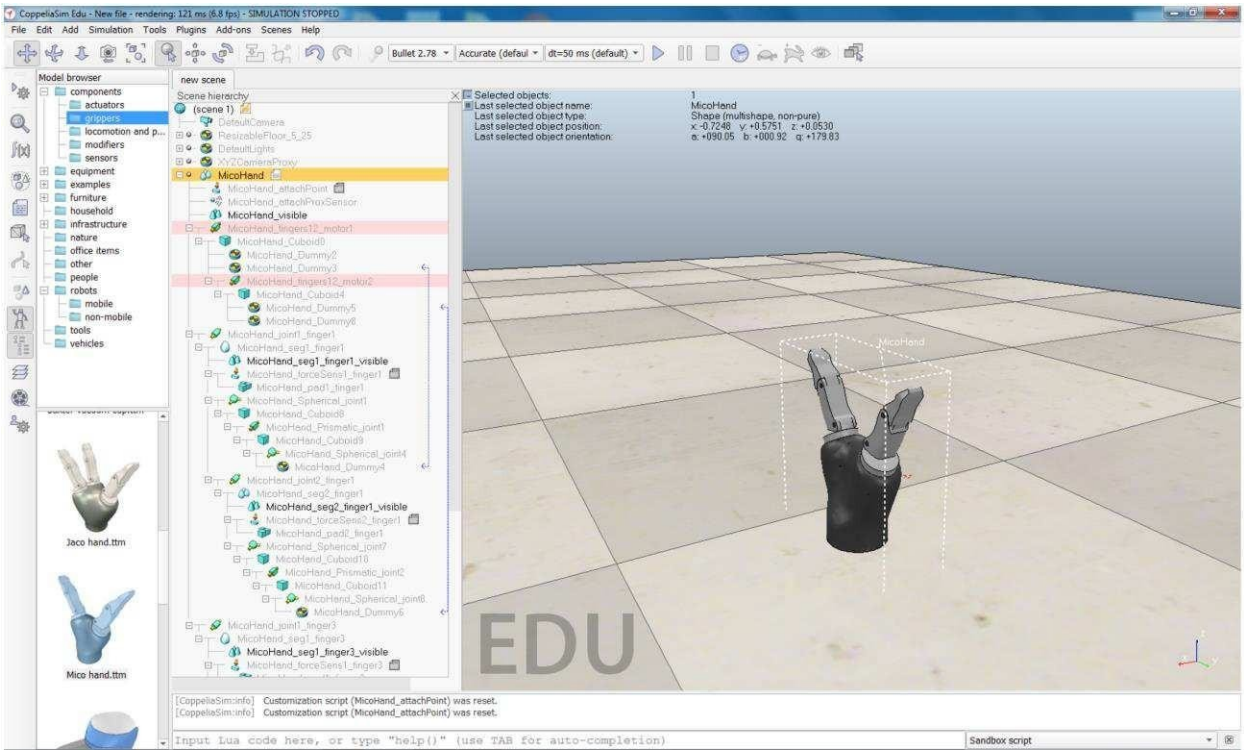

Figure 4.15 MICO robotic arm gripper in CoppeliaSim environment

The complete environment is a combination of all the above-mentioned components. Figure 4.16 visualizes the complete environment. As can be seen in the figure, the robot is placed on the edge of a chair. A human model is also placed on the chair. The robot can move from the table to the mouth of the human model. The three previously mentioned cups with red, green, and blue colors on a table are placed in front of the user. The table, chair, and plant can be easily found in the components as well.

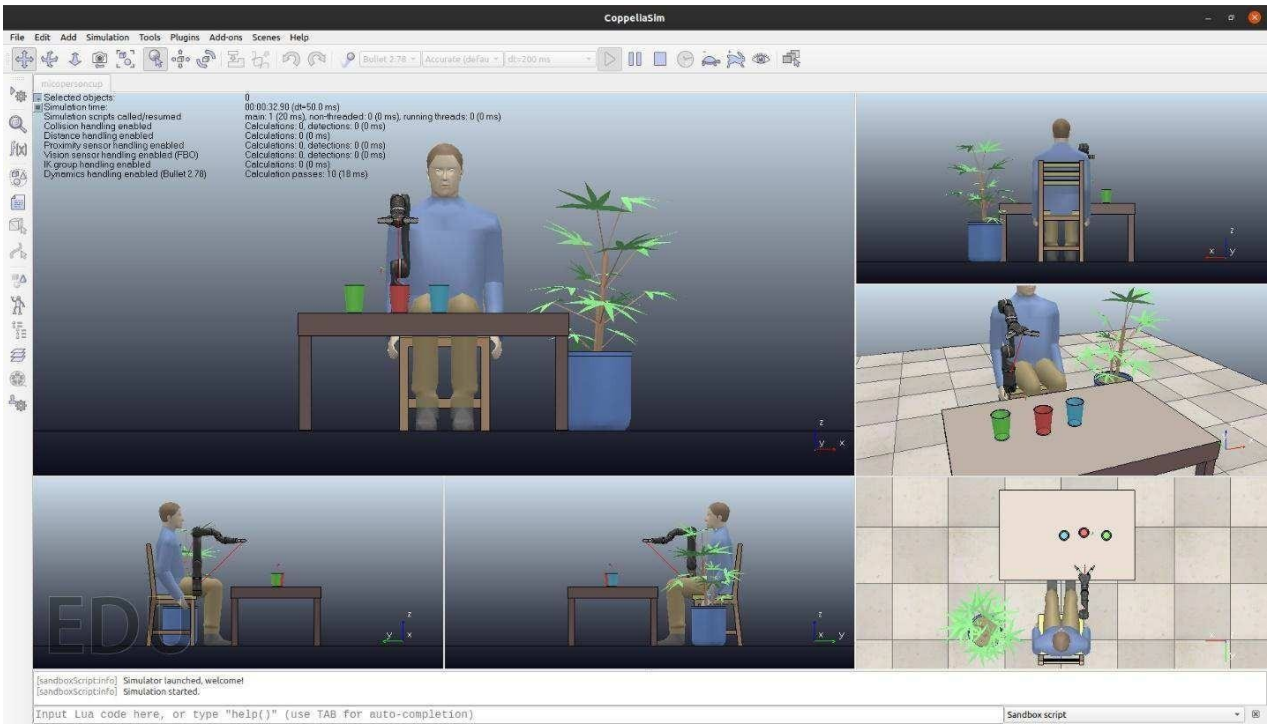

Figure 4.16 The complete overview of simulation

4.8.3 Integration of Algorithm and Simulation

Finally, the proposed algorithm is linked to the CoppeliaSim environment in Linux using the client/server connection, and the robot is actuated with the direct command of angle for each joint. The server sends the control commands to the CoppeliaSim environment. Each command consists of several tasks. For example, grasping a new cup happens in this order:

1. Move the arm above the last selected cup.
2. Move it above the new cup.
3. Grasp the desired cup.
4. Move the cup near the mouth.
5. Wait.
6. Bring the cup to its initial position.
7. Open the gripper.
8. Rest.

For each task, a pre-defined angle for each joint is applied to them. The joints move at a constant pre-defined speed. The client then receives the commands and performs them in the simulation environment. Figures (4.16, 4.17 and 4.18) show the results of the chain of the action.

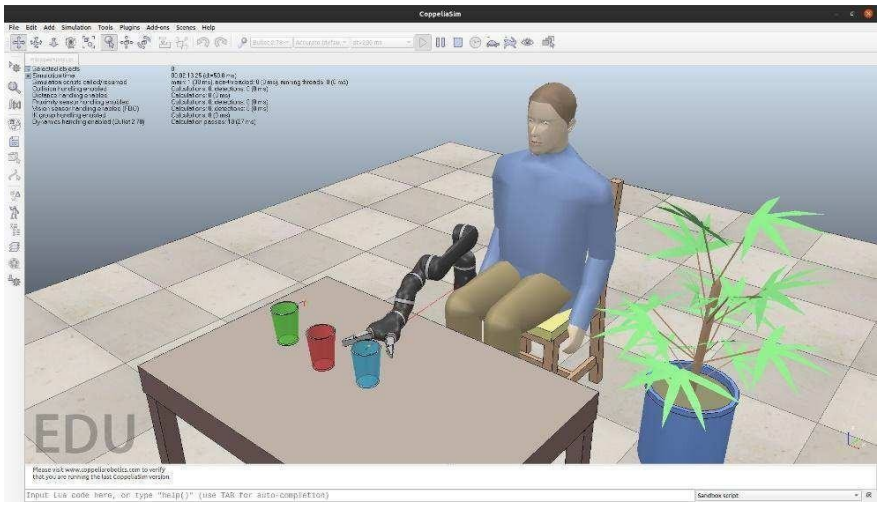

Figure 4.17 The sequence of choosing a cup

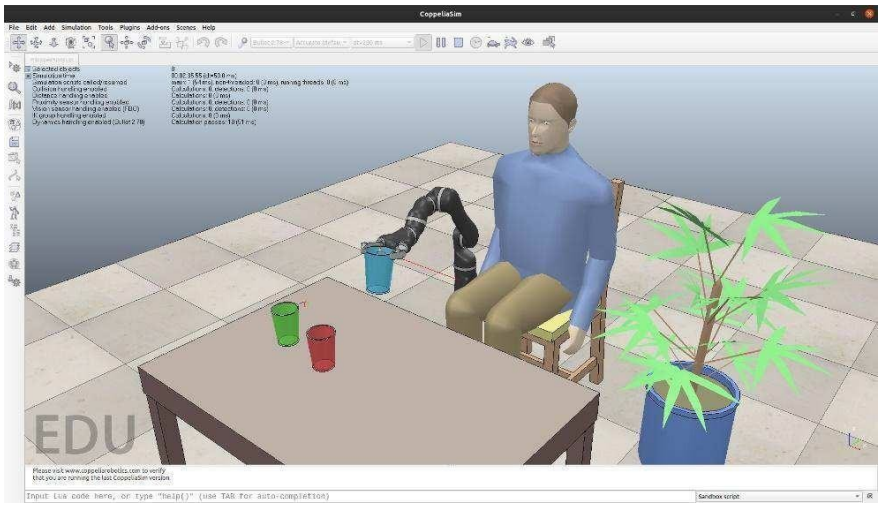

Figure 4.18 The sequence of picking up a cup

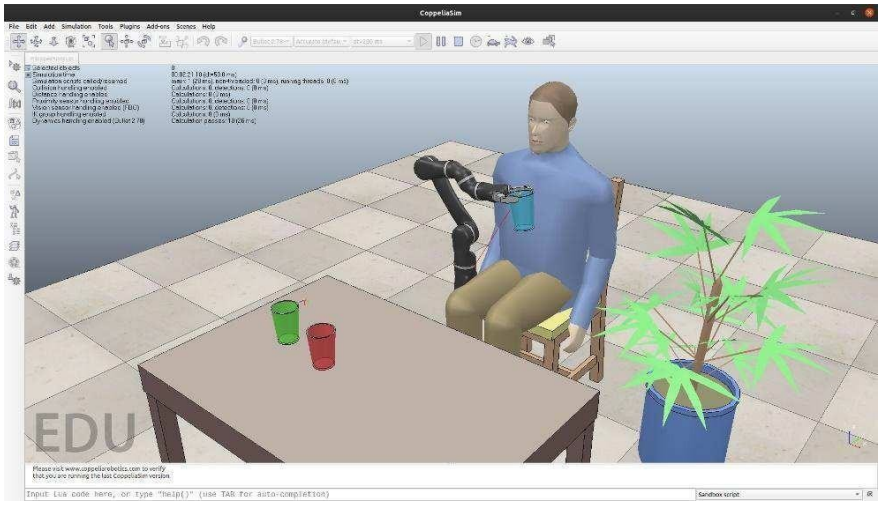

Figure 4.19 The sequence of bringing a cup to a user

CHAPTER 5 RESULTS AND DISCUSSIONS

This chapter aims to illustrate the outputs and results of the proposed method developed in this paper.

5.1 Evaluation Results

Even though the implemented face recognition model has shown that it can function appropriately and become integrated with the arm, it is still necessary to evaluate its performance. As discussed previously, the model is used to recognize the user's face and determine the opening or closing of the mouth or eyes. Therefore, the model's performance with regard to opening and closing is evaluated in this section.

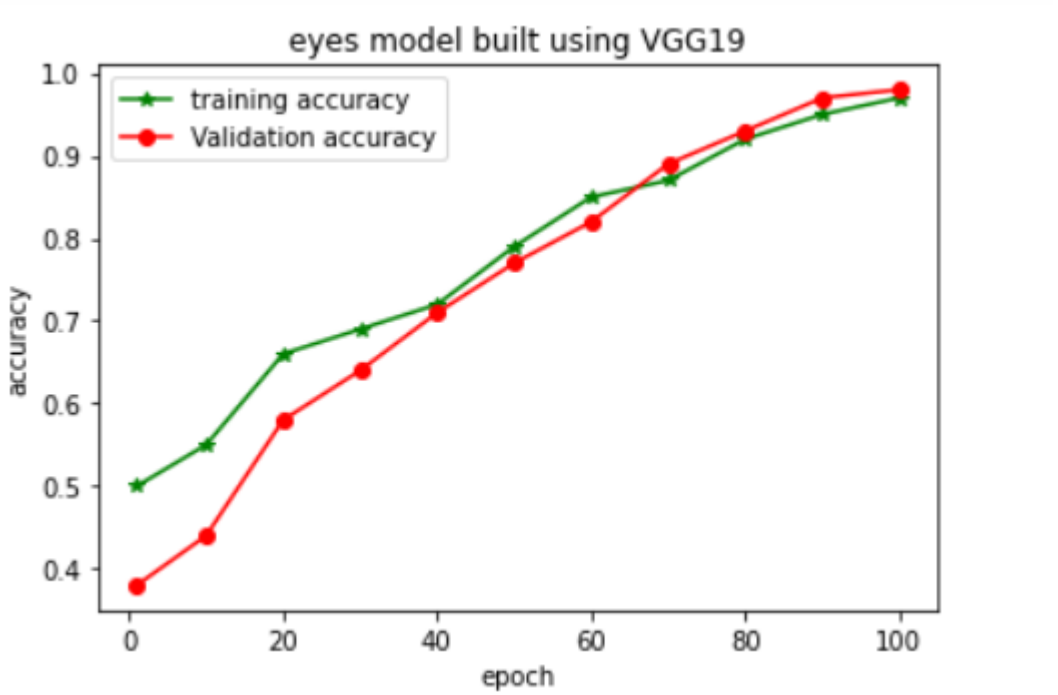

Figure 5.1 Model's accuracy in detecting eyes

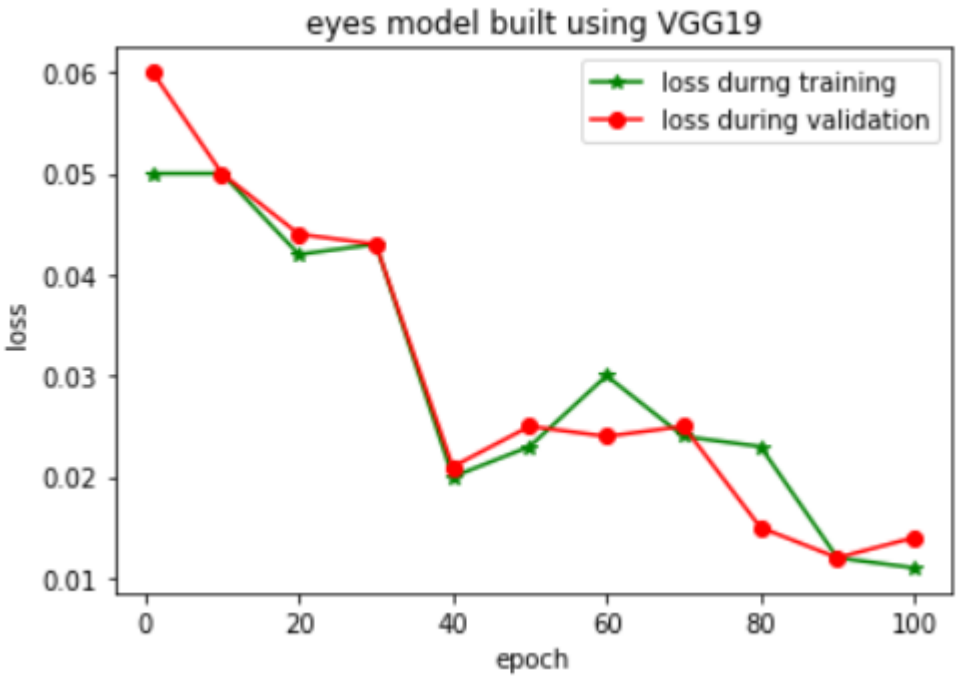

Figure 5.2 Loss experienced by model in detecting eyes

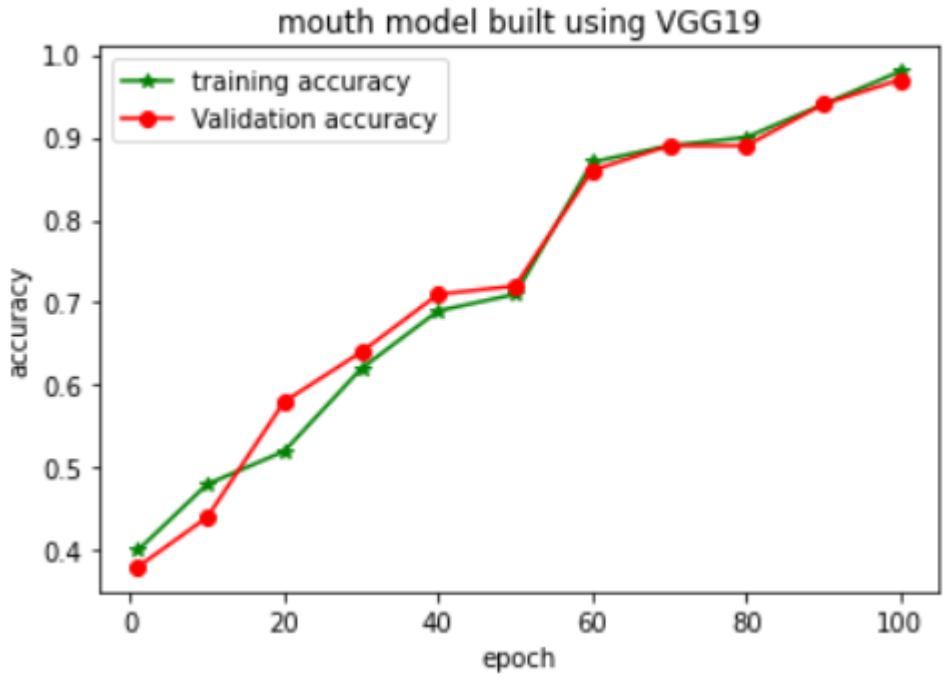

Figure 5.3 Model's accuracy in detecting mouth

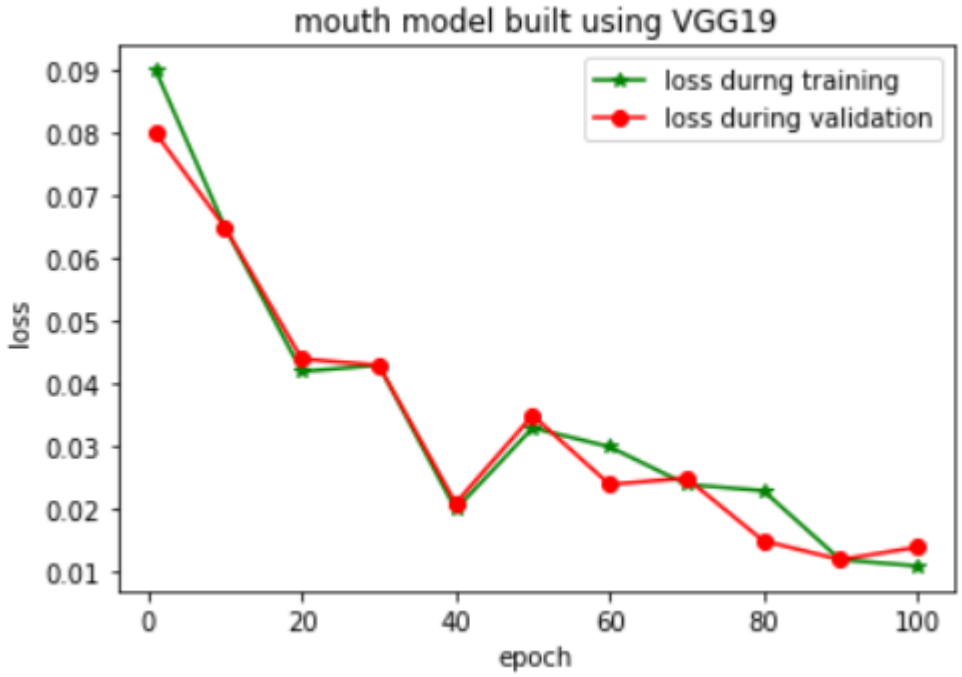

Figure 5.4 Loss experienced by model in detecting mouth

The performance of the model in detecting the movement of eyes and mouth is shown in Figures 5.1 to 5.4, depicting the loss and accuracy at each epoch. Accuracy is one of the metrics used to assess the performance of the model. Separate models are built for predicting the movement of eyes and mouth. Accuracy represents the number of correct predictions made by the model, which takes into consideration the true positive (TP) and false positive (FP) values predicted by the model in relation to the total number of data points in the dataset. The confusion matrix of the model, shown in Figure 5.5, provides TP, FP, true negative (TN), and false negative (FN) values. TP and TN represent correct predictions, while FP and FN represent incorrect predictions.

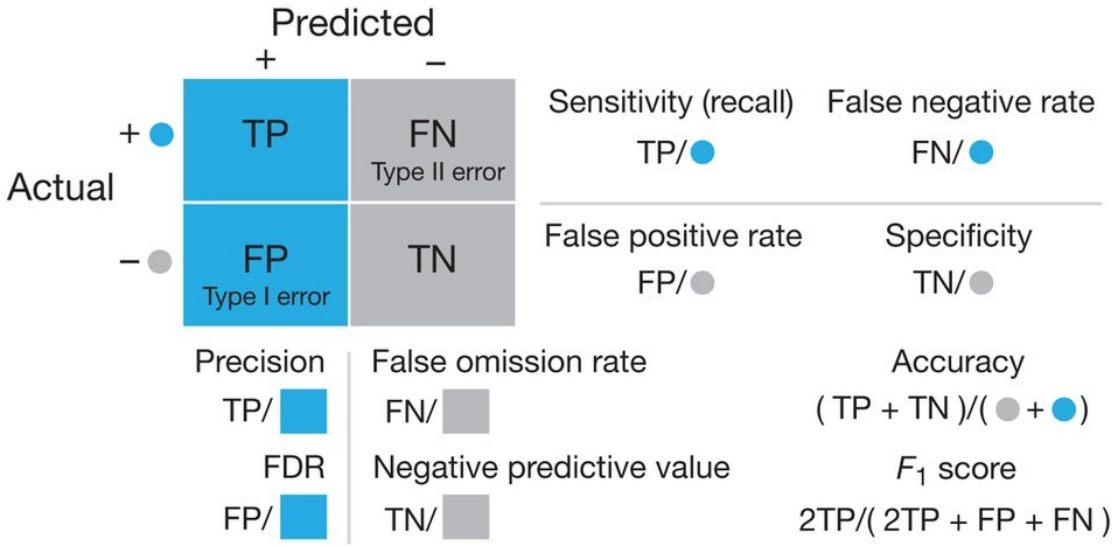

Figure 5.5 Performance metrics of deep learning model [55]

Loss is another metric used to evaluate the performance of the model. A lower loss value indicates better performance. Figure 5.2 shows the loss values at every 10 epochs during the training process, as the model was trained for 100 epochs. Similarly, Figures 5.3 and 5.4 depict the accuracy and loss values for the mouth model, respectively.

The accuracy of the eye model is 98%, while the accuracy of the mouth model is 97%. This means that the model is able to make correct predictions for eye movement 98% of the time and for mouth movement 97% of the time. It's important to note that accuracy and loss are not directly correlated and do not necessarily show any trends together. The key takeaway is that a lower loss value indicates better accuracy of predictions.

The Confusion Matrix of the eye and mouth models are shown in Figures 5.6 and 5.7, respectively. These matrices illustrate how well the model predicted the open or closed state of eyes and mouth. In the Confusion Matrix, there are two different states: open or closed (0 and 1), which represent the real-data and predicted data states. These matrices provide insights into the accuracy of the model's predictions for different states.

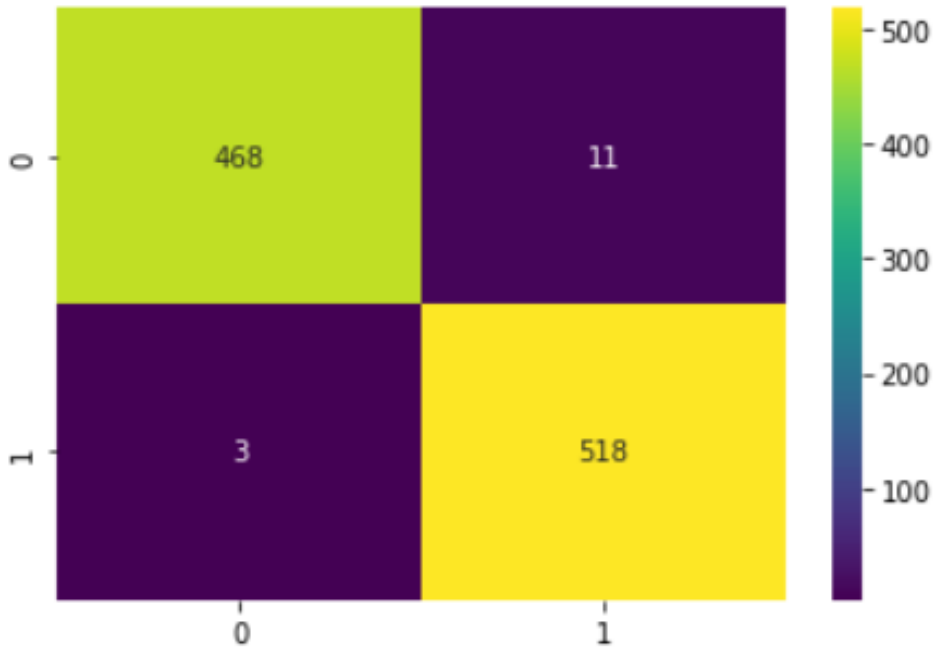

Figure 5.6 Eyes' confusion matrix

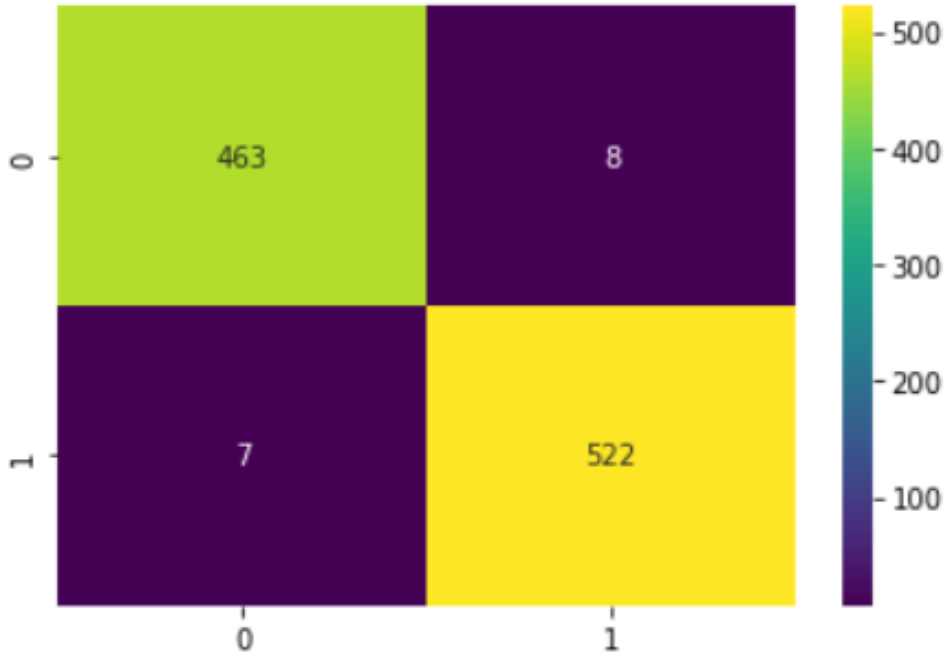

Figure 5.7 Mouth's confusion matrix

The conclusions drawn from above graphs that model is having good accuracy. It is trained with 10000 training images and 4000 testing images and having an accuracy of 98% and 97% means that model is able to classify the eyes into open or close category and also for mouth without overfitting. The loss values are less which means that there is very less ambiguity involved in the data provided to train because of which it is having high accuracy. The model's performance is considered across the deep learning algorithm performance metrics like accuracy and loss with the help of confusion matrix.

CHAPTER 6 CONCLUSION AND FUTURE WORK

Deep-learning algorithms have become an essential part of various industries, ranging from medical products to unmanned aircraft. In the field of mobility assistive devices, deep learning has emerged as the state-of-the-art approach for assistive robotic arms, offering improved accuracy, speed, and capabilities to aid users. Classical algorithms often have inherent limitations, making deep learning increasingly preferred among scientists as an incremental tool.

In recent years, many studies have explored different neural networks for various purposes. In this work, a convolutional neural network (CNN) was employed to recognize the user's face and enable the user to command various tasks through eye and lip movements. CNNs are renowned for their imaging and video processing capabilities, as they excel at extracting essential features. The implemented CNN in this study processes input data from the installed camera on the assistive arm, and after feature extraction, the information is integrated into the mechanical arms to follow the user's commands.

Since an assistive robotic arm like MICO needs to support various user demands, it is crucial to develop novel features that make the arm more user-friendly. Accordingly, this project aimed to develop an algorithm to extract relevant facial features and integrate them into the MICO arm in a simulation platform. This approach provides the user with an assistive arm that is controlled by their facial movements, with eye and mouth movements used to manipulate the robot to choose between different-colored cups to be placed in front of the user's mouth. A graphical simulation environment was designed, and a novel adaptive facial recognition method was implemented to account for face angle and blinking noises. The final results align with the goals of the thesis, with the face recognition model achieving 98% accuracy for eye movements and 97% for mouth movements, validating its reliability for real-world implementation and integration with a robotic arm.

A significant contribution of this project is the implementation of deep learning algorithms that enable the robot to make decisions autonomously and provide assistance to people with disabilities. The flexibility of integrating new features and other deep learning algorithms with the hardware in this setup is a notable advantage. In future work, additional features could be added, and new deep learning algorithms could be integrated easily. The integration of deep learning algorithms with the hardware in this setup is a novel aspect of this project.

As a continuation of this thesis work, there are several potential improvements that could be explored. One compelling idea is to develop a voice control-based tool for the MICO arm using signal processing techniques. Additionally, improving the precision of the robot's movements and increasing its angles and degrees of freedom could enhance its ability to serve users with mobility impairments.

REFERENCES

[1] Berrigan, P., Scott, C. W., & Zwicker, J. D. (2020). Employment, education, and income for Canadians with developmental disability: Analysis from the 2017 Canadian Survey on Disability. Journal of Autism and Developmental Disorders, 1-13.

[2] Vogel, J., Haddadin, S., Jarosiewicz, B., Simeral, J. D., Bacher, D., Hochberg, L. R., ... & van der Smagt, P. (2015). An assistive decision-and-control architecture for force-sensitive hand–arm systems driven by human–machine interfaces. The International Journal of Robotics Research, 34(6), 763-780.

[3] Assistive Robotics Market Global Forecast to 2024. https://www.marketsandmarkets.com

[4] http://eecs.ucf.edu/~abehal/AssistiveRobotics/presentation/UCF_MANUS_ARM_082009.pdf

[5] K. Inc., "Kinova MICO Robotic Arm User Guide," 2018. [En ligne]. Disponible: https://drive.google.com/file/d/1x9hn4ODrroSgCyQMzJclIGWOkiOwOhbC/view

[6] K. Inc., "Kinova JACO Robotic Arm User Guide,". [En ligne]. Disponible: https://assistive.kinovarobotics.com/uploads/EN-UG-007-Jaco-user-guide-R05.pdf

[7] Driessen, B. J., Kate, T. T., Liefhebber, F., Versluis, A. H. G., & Woerden, J. V. (2005). Collaborative control of the manus manipulator. Universal Access in the Information Society, 4, 165-173.

[8] Flandin, G., Chaumette, F., & Marchand, E. (2000, April). Eye-in-hand/eye-to-hand cooperation for visual servoing. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065) (Vol. 3, pp. 2741-2746). IEEE.

[9] Tsui, K., Yanco, H., Kontak, D., & Beliveau, L. (2008, March). Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction (pp. 105-112).

[10] Bousquet-Jette, C., Achiche, S., Beaini, D., Cio, Y. L. K., Leblond-Ménard, C., & Raison, M. (2017). Fast scene analysis using vision and artificial intelligence for object

prehension by an assistive robot. Engineering Applications of Artificial Intelligence, 63, 33-44.

[11] Schröer, S., Killmann, I., Frank, B., Völker, M., Fiederer, L., Ball, T., & Burgard, W. (2015, May). An autonomous robotic assistant for drinking. In 2015 IEEE international conference on robotics and automation (ICRA) (pp. 6482-6487).

[12] Lampe, T., Fiederer, L. D., Voelker, M., Knorr, A., Riedmiller, M., & Ball, T. (2014, February). A brain-computer interface for high-level remote control of an autonomous, reinforcement-learning-based robotic system for reaching and grasping. In Proceedings of the 19th international conference on Intelligent User Interfaces (pp. 83-88).

[13] Philis, J., Millán, J. D. R., Vanacker, G., Lew, E., Galán, F., Ferrez, P. W., ... & Nuttin, M. (2007, June). Adaptive shared control of a brain-actuated simulated wheelchair. In 2007 IEEE 10th International Conference on Rehabilitation Robotics (pp. 408-414).

[14] Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., ... & Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature, 485(7398), 372-375.

[15] Tapia, E. M., Intille, S. S., & Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. In Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings 2 (pp. 158-175). Springer Berlin Heidelberg.

[16] Chen, O. T. C., Tsai, C. H., Manh, H. H., & Lai, W. C. (2017, August). Activity recognition using a panoramic camera for homecare. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6).

[17] Zhang, Z., Huang, Y., Chen, S., Qu, J., Pan, X., Yu, T., & Li, Y. (2017). An intention-driven semi-autonomous intelligent robotic system for drinking. Frontiers in neurorobotics, 11, 48.

[18] Leroux, M., Raison, M., Adadja, T., & Achiche, S. (2015, May). Combination of eyetracking and computer vision for robotics control. In 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA) (pp. 1-6). IEEE.

[19] Tanaka, H., Sumi, Y., & Matsumoto, Y. (2010, October). Assistive robotic arm autonomously bringing a cup to the mouth by face recognition. In 2010 IEEE workshop on advanced robotics and its social impacts (pp. 34-39). IEEE.

[20] Zhang, Y., Hornfeck, K., & Lee, K. (2013). Adaptive face recognition for low-cost, embedded human-robot interaction. In Intelligent Autonomous Systems 12: Volume 1 Proceedings of the 12th International Conference IAS-12, held June 26-29, 2012, Jeju Island, Korea (pp. 863-872). Springer Berlin Heidelberg.

[21] Aronson, R. M., Santini, T., Kübler, T. C., Kasneci, E., Srinivasa, S., & Admoni, H. (2018, February). Eye-hand behavior in human-robot shared manipulation. In Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (pp. 4-13).

[22] Cio, Y. S. L. K., Raison, M., Menard, C. L., & Achiche, S. (2019). Proof of concept of an assistive robotic arm control using artificial stereovision and eye-tracking. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(12), 2344-2352.

[23] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10), 1499-1503.

[24] Bengtson, S. H., Bak, T., Andreasen Struijk, L. N., & Moeslund, T. B. (2020). A review of computer vision for semi-autonomous control of assistive robotic manipulators (ARMs). Disability and Rehabilitation: Assistive Technology, 15(7), 731-745.

[25] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[26] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[27] Qassim, H., Verma, A., & Feinzimer, D. (2018, January). Compressed residual-VGG16 CNN model for big data places image recognition. In 2018 IEEE 8th annual computing and communication workshop and conference (CCWC) (pp. 169-175). IEEE.

[28] Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441.

[29] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[30] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustainable cities and society, 66, 102692.

[31] Li, Z., & Zhou, F. (2017). FSSD: feature fusion single shot multibox detector. arXiv preprint arXiv:1712.00960.

[32] Jiang, B., Luo, R., Mao, J., Xiao, T., & Jiang, Y. (2018). Acquisition of localization confidence for accurate object detection. In Proceedings of the European conference on computer vision (ECCV) (pp. 784-799).

[33] Coste, A. (2012). Image Processing: Affine Transformation. Landmarks registration, Nonlinear Warping.

[34] Kaur, P., Krishan, K., Sharma, S. K., & Kanchan, T. (2020). Facial-recognition algorithms: A literature review. Medicine, Science and the Law, 60(2), 131-139.

[35] Cech, J., & Soukupova, T. (2016). Real-time eye blink detection using facial landmarks. Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague, 1-8.

[36] Galindo, R., Aguilar, W. G., & Reyes Ch, R. P. (2019). Landmark based eye ratio estimation for driver fatigue detection. In Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part V 12 (pp. 565-576). Springer International Publishing.

[37] Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In 2018 IEEE international workshop on information forensics and security (WIFS) (pp. 1-7). IEEE.

[38] Reghunath, A., Nair, S. V., & Shah, J. (2019, July). Deep learning based customized model for features extraction. In 2019 International Conference on Communication and Electronics Systems (ICCES) (pp. 1406-1411). IEEE.

[39] Cheng, S., & Zhou, G. (2020). Facial expression recognition method based on improved VGG convolutional neural network. International Journal of Pattern Recognition and Artificial Intelligence, 34(07), 2056003.

[40] Gong, W., Chen, H., Zhang, Z., Zhang, M., Wang, R., Guan, C., & Wang, Q. (2019). A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. Sensors, 19(7), 1693.

[41] Wang, M., Lu, S., Zhu, D., Lin, J., & Wang, Z. (2018, October). A high-speed and low-complexity architecture for softmax function in deep learning. In 2018 IEEE asia pacific conference on circuits and systems (APCCAS) (pp. 223-226). IEEE.

[42] Khattar, A., & Quadri, S. M. K. (2022). Generalization of convolutional network to domain adaptation network for classification of disaster images on twitter. Multimedia Tools and Applications, 81(21), 30437-30464.

[43] Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018, April). An end-to-end deep learning architecture for graph classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[44] Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 54(10), 6232-6251.

[45] Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25, 197-227.

[46] Lin, W., Wu, Z., Lin, L., Wen, A., & Li, J. (2017). An ensemble random forest algorithm for insurance big data analysis. Ieee access, 5, 16568-16575.

[47] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. Mechanical systems and signal processing, 151, 107398.

[48] Stathakis, D. (2009). How many hidden layers and nodes?. International Journal of Remote Sensing, 30(8), 2133-2147.

[49] Guo, G., & Zhang, N. (2019). A survey on deep learning-based face recognition. Computer vision and image understanding, 189, 102805.

[50] Li, G., & Yu, Y. (2018). Contrast-oriented deep neural networks for salient object detection. IEEE transactions on neural networks and learning systems, 29(12), 6038-6051.

[51] Ren, L., Lu, J., Feng, J., & Zhou, J. (2019). Uniform and variational deep learning for RGB-D object recognition and person re-identification. IEEE Transactions on Image Processing, 28(10), 4970-4983.

[52] Coppelia Robotics (Blue Workforce) is a provider of Industrial IoT robots' technologies based in Switzerland. Available in: https://www.coppeliarobotics.com/

[53] PyRep is a toolkit for robot learning research, built on top of CoppeliaSim (previously called V-REP). Install; Running Headless; Getting Started; available in: https://github.com/stepjam/PyRep

[54] James, S., Freese, M., & Davison, A. J. (2019). Pyrep: Bringing v-rep to deep robot learning. arXiv preprint arXiv:1906.11176.

[55] Nature Methods https://www.nature.com/articles/nmeth.3945#Fig1

APPENDIX A MODEL ARCHITECTURE

Here is the description of the architecture of the model. This section aims to provide more details about the model.

- An RGB (Red Green Blue) input image of size 224*224 is given as input to this architecture.
- The kernel size is 3*3, and the stride size is 1 pixel. This helps the network to cover the entire image.
- In order to reduce spatial resolution, passing is implemented.
- Pooling was implemented for 2*2 pixel, with stride size equal to 1 pixel.
- Relu was the activation function, as it has a much faster computation time and therefore is better suited for classification problems.
- This architecture contains three fully connected layers that deal with input image of 4,096, along with many channels.