Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Eye based wheelchair guidance

July 2021

Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Eye based wheelchair guidance

This documentation submitted as required for the degree of bachelors
in Computer and Information Sciences.

By

Ahmad Mohammad Abdel-Hafeez [Scientific Computing Department]

Ayaalla Mohammad ElTabey [Scientific Computing Department]

Ahmad Osama Mohammad [Scientific Computing Department]

Bassant Ehab Moustafa [Scientific Computing Department]

Asmaa Ali El-Sheikh [Scientific Computing Department]

Under Supervision of

[Dr/ Manal Mohsen Tantawi]
[Associate Professor at Scientific Computing Department],
Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

[TA/ Radwa Reda Hossieny]
[Teaching Assistant at Scientific Computing Department],
Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgements

All praise and thanks to ALLAH, who provided us the ability to complete this work. We hope to accept this work from us.
We are grateful of our parents and family who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

I also offer my sincerest gratitude to my supervisors, Associate *Prof. Dr.Manal Mohsen Tantawi, and T.A. Radwa Reda Hossieny* who have supported me throughout my thesis with their patience, knowledge and experience.

Finally, I would thank my friends and all people who gave me support and encouragement.

Abstract

Human Computer interface (HCI) systems are a hot spot the recent researches where are playing a vital role in advancing the quality of life for disable people. It has become a way for people with disabilities to communicate with their external environment using biological signals such as Electro-oculogram (EOG).
These interfaces are based on detecting eye movements. Electro-oculogram (EOG) is an electro-physiological signal generated by eye movement and it can be measured using electrodes placed around the eyes. HCI uses such signals to control virtual keyboard, wheel chair, etc.

In this study, we propose a wheelchair guidance system based on EOG signals, Electro-oculogram (EOG) signals are measured using electrodes placed around the eyes six electrodes including the ground electrodes are placed on the forehead. The two channels are arranged vertically and horizontally. The effect of noise in EOG signals is eliminated especially, eye blink artifacts by band pass filter. Signals are classified into six classes: left, right, up, down, center and double blinking using deep learning models as Convolution neural network(CNN), INCEPTION-V1, INCEPTION-V2, INCEPTION-V3, VGG16, VGG19, RESNET and RESNET-50 which achieved the best average accuracy 95.83%. Finally, these movements are used to send a command to the wheelchair by the direction of movement. Our goal is to achieve high interaction speed in real time.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

- **EOG:** Electrooculogram
- **KNN**: K-Nearest Neighbor
- **SVM**: Support Vector Machine
- **CNN**: Convolution Neural Network
- **ReLU**: Rectified Linear Unit
- **ResNet**: Residual Network
- **VGG**: Visual Geometry Group

Chapter 1: Introduction

1.1 Problem Definition

1.2 Motivation

1.3 Objectives

1.4 Time plan

In this chapter, we will present an overview on the topic and explain the problem and the motives that made us apply to do this project that provide services that benefit many people and also we clarify the goal that we sought to achieve, and in the end we will display the document outlines to explain briefly the content of rest of the chapters.

1.1 Problem definition:

One of the main issues that we are aiming to help in the society are those of the disabled. Disabilities does not have one single type or manner that it attacks the body but comes in a very wide range. The disabilities that we are trying to aid are those who are unable to move the standard wheels or joystick i.e. fully paralyzed people. People with disabilities need to feel some sort of independence; therefore, we thought of implementing the use of eyesight to mobilize their wheelchairs as a way to overcome their disability and provide them with the freedom they yearn for.
In the European Union there are about 80 million elderly or disabled people.

Various reports also show that there is a strong relation between the age of the person and the handicaps suffered, the latter being commoner in persons of advanced age. Given the growth in life expectancy in the EU, this means that a large part of its population will experience functional problems. Aware of the dearth of applications for this sector of the population, governments and public institutions have been promoting research in this line in these recent years. Various types of research groups at a world level have begun to set up cooperation projects, projects to aid communication and mobility of elderly and/or disabled persons with the aim of increasing their quality of life and allowing them a more autonomous and independent lifestyle and greater chances of social integration.

In recent years there has been an increase in the development of assistive technology for people with several disabilities, and great strides have been made in communication systems between humans and machines. These advances have been made mainly in communications from machines to humans, by means of graphical user interfaces or multimedia applications (sounds). However, these advances in the communication from humans to machines have been modest, using keyboards, mice, joysticks, or tactile screens.

All these systems are manual. At the moment, many communication systems are being developed based on voice recognition or visual information and they will be launched onto the market in the coming years.

For example, people daily make a lot of eye movements that allow them to do different tasks, such as reading, writing, learning new things, acquiring information about the environment, handling objects and communicating with other people, etc. This ability of people to control their eye direction can be used to communicate with the machines.

Electrical wheelchair control: In addition to the virtual keyboards, we evaluated the usability of forehead EOG for driving a power wheelchair. In view of safety considerations, experienced subjects participated in the experiment.
The commercialized power wheelchair was modified to drive using forehead EOG signals. The EOG signals measured by the headband sensor were filtered, amplified, and transmitted to a tablet PC wirelessly via Bluetooth. Then, a classification program classified the eye movements and transmitted the results to a custom-made digital–analog converting module that replaced the wheelchair's joystick controller. Four types of eye movements were used as commands to change the state of the wheelchair. the power wheelchair is shown in figure 1.1.

Fig. 1.1: Wheelchair driving using wearable forehead EOG sensor

Electro-oculogram (EOG) signals

There are also many applications based on the human computer interface such as: mobile robot control, multitask gadget control, computer cursor control, computer animation application, hospital alarm system, activity recognition based on eye movement analysis, visual improvement system for the elderly, home automation and Games.

An Electro-oculogram (EOG) is a bio-potential signal that measures the potential difference between the retina and the cornea. The eyeball can be modeled as a dipole with the positive cornea in the front and the negative retina in the back, positive or negative pulses will be generated when the eyes roll upward or downward. The amplitude of pulse will be increased with the increment of rolling angle, and the width of the positive (negative) pulse is proportional to the
duration of the eyeball rolling process. Simultaneously, the EOG is recorded in the two directions horizontal and vertical using five electrodes (two vertical, two horizontal and one for ground) placed around the eyes.

Two channel EOG signals, horizontal and vertical signals, have been commonly used to acquire information from human eye movements. Independent measurements can be obtained from both eyes. However, in vertical directional movements, two eyes move in conjunction, thus only one right eye was deployed. The procedure of recorded EOG signals is presented in the following.
Five surface electrodes were placed around the eyes. All positions are as shown in Figure 4. Vertical-channel electrodes were placed above and below the right eye (Ch.V+ and Ch.V-) and horizontal channel electrodes were placed on the right and left of the outer canthi (Ch.H+ and Ch.H-). Additionally, a reference electrode was placed on the forehead (G).

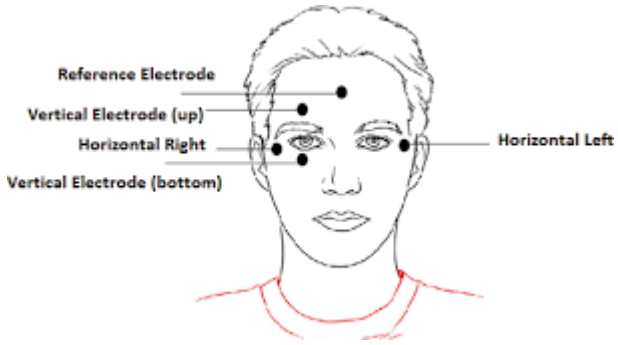

Fig. 1.2: Electrode placement for conventional EOG

We can measure the EOG signal for 6 eye movements (Up, Down, Right, Left, Blinking and No movement) as shown in Figures 5 and 6.

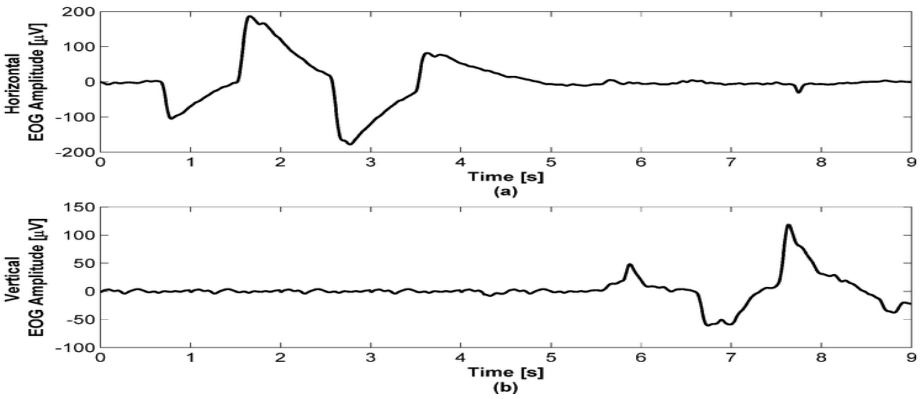

Fig. 1.3: EOG signals for eye movements Horizontal, Vertical

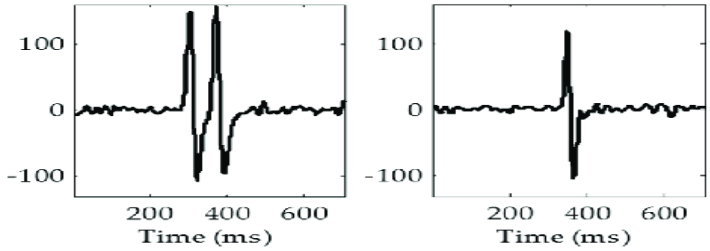

Fig. 1.4: Forehead EOG signal pattern

1.2 Motivation:

One of the most potentially useful applications for increasing the mobility of disabled and/or elderly persons [In 2018, in Egypt 2.5% of the population suffers from disability and they are in increasable as the most likely to reach 12 million disabled people] is wheelchair implementation. A standard motorized wheelchair aids the mobility of disabled people who cannot walk, always providing that their disability allows them to control the joystick safely. Persons with a serious disability or handicap, however, may find it difficult or impossible to use them; cases in point could be tetraplegics who are capable only of handling an on–off sensor or make certain very limited movements, such as eye movements. So, one of the main motives behind this research work has therefore been the aim of making a contribution towards satisfying the technological needs of potential wheelchair users by designing an eye-movement guidance system for severely disabled persons, with an economic and functional feasibility that would enable them to improve their quality of life.

1.3 objective:

The objective of this project is to develop a wheelchair guidance system based on electrooculography (EOG).

1.4 Time plan:

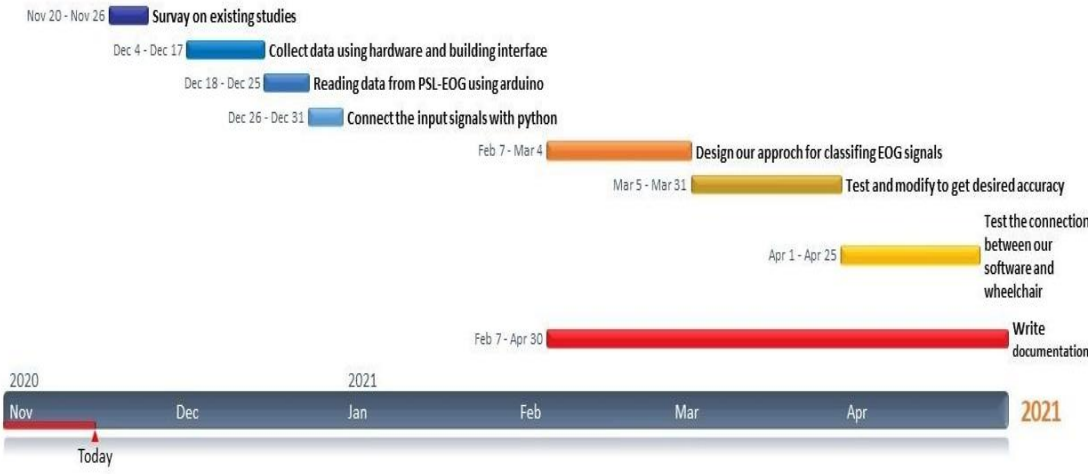

Fig. 1.5: Time plan

1.5 Documentation Outline:

The rest of the document is divided into six main Chapters and a separate Chapter for conclusions and future work at the end of the document.

Chapter 2: presents a detailed overview on the main concepts and theory related to the electro-oculography and controlling wheelchair using it. It gives summary for the history developed applications which use the technology of EOG.

Chapter 3: reviews the system architecture and description of each part and most of the recent and famous algorithms that use to apply each one.

Chapters 4: The proposed work of the document is presented in.

Chapter 5: present the process to test our system and result of it.

Chapter 6: Give a user manual to handle the system.

Chapter 7: conclusion chapter.

Chapter 2: Background

2.1 Related work

2.1.1 Threshold method

2.1.2 Traditional methods

2.1.3 Deep learning methods

2.2 Observations

Before starting in any project or in any topic, we must see what researchers have done. Then it's important to make survey on papers which deal with our interests to see what techniques and Methods that the researchers used to deal with our project or our topic, so we have read many papers that deal with EOG signal. [1-27]
In this chapter we discuss some papers by following general system architecture and we show the parts pre-processing, feature extraction, classification and results that mentioned in each paper. for each paper we show what filters were used, what features were extracted, what classification algorithm was applied and what accuracy and time that the researchers got in paper.

2.1 related work:

2.1.1 Using Threshold method:

Yamagishi and et al. [1] discussed how the limited number of eye movements is the main reason for the slow performance of EOG based Eye typing systems and proposed combining 4 directions to the basic 4 directions, namely: Up, down, right, and left combined with: Up right, up left, down right and down left. The selection was also using intentional eye blinks as a mouse click. The EOG was measured with 2 electrodes, the proposed 8 directional cursor movements were extracted from logical comparisons of both channel values versus threshold settings tailored to each individual according to corresponding trials. In their results, they reported that using 8-directional cursor movements instead of 4, increased the typing speed from 10 to 12 letters per minute while the accuracy of writing remained the same, almost 90.4%.

Jeong Heo and et al. [2] proposed a new practical electrode position on the forehead to measure EOG signals, used Low-Pass Filter with 10 Hz cutoff to reduce noise, the maximum and minimum values of the peak and their time indexes used to capture features, and threshold as classifier, with accuracy 91%.

Manuel Merino Monge and et al. [3], detected the direction of eye movements, they used bans pass filter with cutoff [0.1,30] Hz to reduce noise, and used tolerance to capture the features, a classification equation was used to classify the signals (if (value)<Amplitude threshold -> tolerance ->low), Where "value" is the absolute Amplitude, the accuracy was 94%.

Rui Zhang and et al. [4], Controlled a Smart Home Environment, using zero-phase filter between 0.1 Hz and 20 Hz to remove high-frequency, the amplitude and duration features from EOG signals for blink detection to capture features,

threshold method is used to classify, the patients issued control commands with an average accuracy of 93.6%.

Kwang-Ryeol Lee and et al. [5] In preprocessing stage researchers use a median filter with a fixed window size was applied to the raw EOG signal, the window size of the median filter was chosen to be 20 points, which was equivalent to 160 milliseconds, the features the researchers extracted from EOG signal were start and end times of eye movements, eye blinks and course directional change in gaze. Then they used fixed values for the slope threshold and the Gaussian function deviation over all participants, and they were empirically set to 0.05 and 8. The accuracy is measured by applying F1 score and they get ranged from 90.14% to 97.81%. The F1 score is the 2*((precision*recall)/ (precision + recall)). It is also called the f1 score or the F Measure. Final This work specifically focused on eye-writing recognition in a stationary environment. Participants sat on a chair and were instructed to stay "naturally motionless" while eye-writing.

2.1.2 Using traditional methods:

Adaresh Rajesh and Megha Mantur and et al. [6] whose develop and work on EOG signals the get their data and split it into 836 for taring and 336 are kept for testing and applied some pre-processing, then they use Hough transformation for feature extraction, and they use Novel deep learning approach by all of that accuracy was 97.95 on training and 99.88% on validation data.

Hema.C.R and et al.[7] , used Electrooculography and Neural Networks to classify eye movements , they used notch filter of 50Hz to reduce noise, used Feature extraction algorithm based on the Parseval and Plancherel theorems are proposed to extract the features from each band, used neural network models classifiers ,with accuracy 91%.

2.1.3 Using Deep Learning methods:

Dr. S. Ramkumar et al. [8], Their data acquisition was on 20 persons only and applied some pre-processing as Chebyshev bandpass and use the Parseval theorem and the Plancherel theorem to extract the feature and for classification they used TDNN and FFNN and get accuracy 92.1925%.

B. Estrany et al. [9], controlled computer by eye movements, they used high pass filter with cutoff 8Hz to reduce the noise, foveal cursor movement patterns are used to capture features, Linear Regression Module used to classify, Accuracy can reach 8 pixels at the resolution of 1024 x 768 and 4 pixels for a screen resolution of 800 x 600 pixels.

2.2 Observations

The datasets which have been used in most of the papers were small dataset only 20 persons.

The most used preprocessing operations were filtering using high pass filter, low pass filter and band pass filter and resampling to 100.

Feature were extracted using geometrical feature such as: Max peak and amplitude position, statistical feature such as: Mean μ_i, Standard deviation, Energy E.

The most used classifiers were threshold methods and using traditional methods such as KNN and FFNN.

The accuracy over all papers doesn't exceed 90%.

Chapter3:System Architecture

3.1 Data Acquisition

3.2 Preprocessing

3.3 Feature Extraction

3.4 Classification

3.5 Controlling Devices

In this chapter we will explain the system architecture for eye-based wheelchair gaudiness system using electrooculography, first of all the system contain many stages, each one has many technique and algorithm that can been applied, so we will describe each stage and it's benefit and the most used and famous algorithms can have been applied.

The system architecture includes five stages: Data acquisition, Pre-processing, Feature extraction, classification and finally controlling Device.

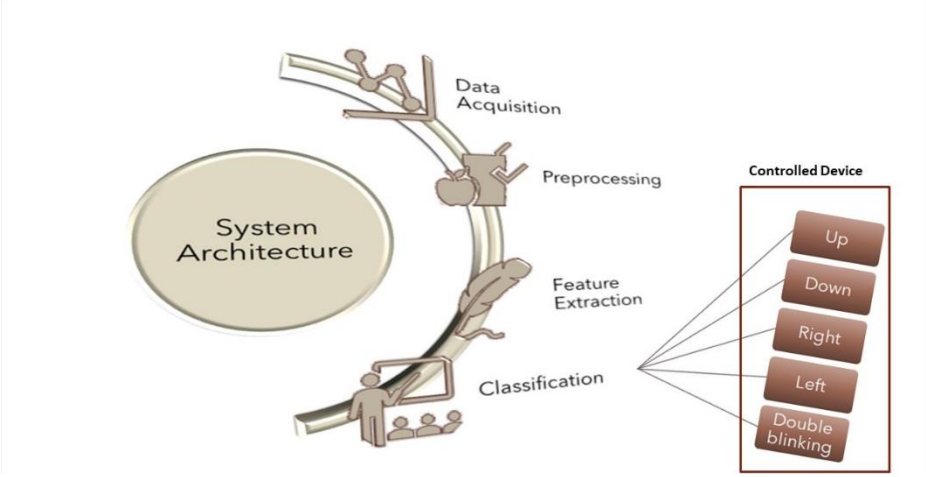

Fig. 3.1: system architecture

3.1 Data Acquisition

In most of the previous EOG-based studies, electrodes were placed such that one was located both above and below the eye (left or right) to measure vertical eye movement and one was located both at the left and right of the outer canthi to measure horizontal eye movement (Figure 10 a). The novel electrode placement proposed in this study is shown in (Figure 10 b).

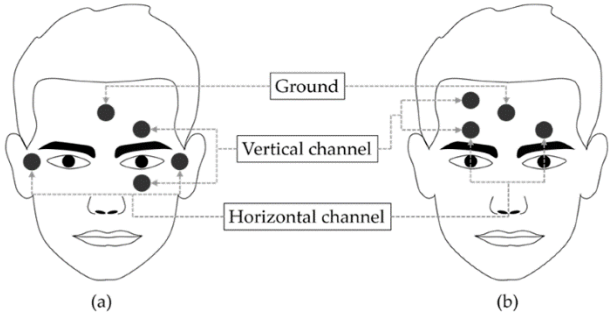

Fig. 3.2: Electrode placement

The positive electrode for the vertical and horizontal channels was shared, and it was placed above the eye. The negative electrode for the vertical channel was placed above the positive electrode.
The negative electrode for the horizontal channel was placed above the other eye. We evaluated the usability of the proposed electrode positioning with a commercial EOG measurement system (EOG100C, Biopac Systems Inc., Goleta, CA, USA). The gain was set to 5000 and the frequency bandwidth was set to 0.05–35 Hz.

3.2 Pre-processing:

After signal acquisition phase the signals are to be pre-processed. In general, the acquired eye signals are contaminated by noise and artifacts, to avoid the eye blinks, small involuntary EOG movements, background noise and power interference.
So, we can use several methods to filter any EOG signal such as:
Static Threshold equal 1/4max amplitude.
Dynamic Threshold to avoid noise from involuntary blinking and variation.
Low pass filter [fifth order butter worse with cutoff=20HZ]
High pass filter [second order butter worse with cutoff=0.05HZ]
Band pass filter [with cutoff band=0.05 -20HZ]
Denise (electrode-off): This is due to the long recording, so we use algorithm use stander deviation if it isn't in [min=0.1nv, max=100Mv] it is not valid for processing until reach to range again.

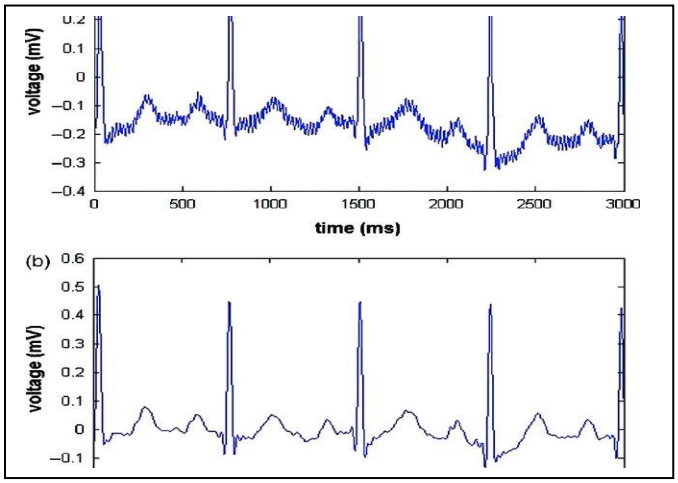

Fig. 3.3: signals before and after filtering

3.3 Feature Extraction:

The extraction feature of the signals will be in the time domain, we can use several methods to extract features from horizontal and vertical EOG signals, they are a combination between some techniques and both channels for EOG signal, all feature are calculated based on time domain in order to yield a computation simplicity, the definition of some techniques has been described in the following.

Statistical Features:

That of the signal to be used in the classification of eye movements, the statistical features are the most commonly used features of biological signals, which help to obtain signal information in the time domain as: mean, stander deviation and energy.

$$1- \quad \mu_i = \sum_{i=1}^{N} X_i$$

$$2- \quad \sigma_i = \sum_{i=1}^{N} (X_i - \mu_i)$$

$$3- \quad E = \sum_{i=1}^{N} X_i^2$$

Geometrical Feature:

That dependent on the shape of the signal and it characteristics as: Maximum Peak amplitude value (PAV), Maximum Valley amplitude value(VAV): PAV It's a measure of the EOG signal amplitude value at the highest point, maximum positive value of peak case and amplitude value at the lowest point, maximum negative value of valley case in both vertical and horizontal channels.

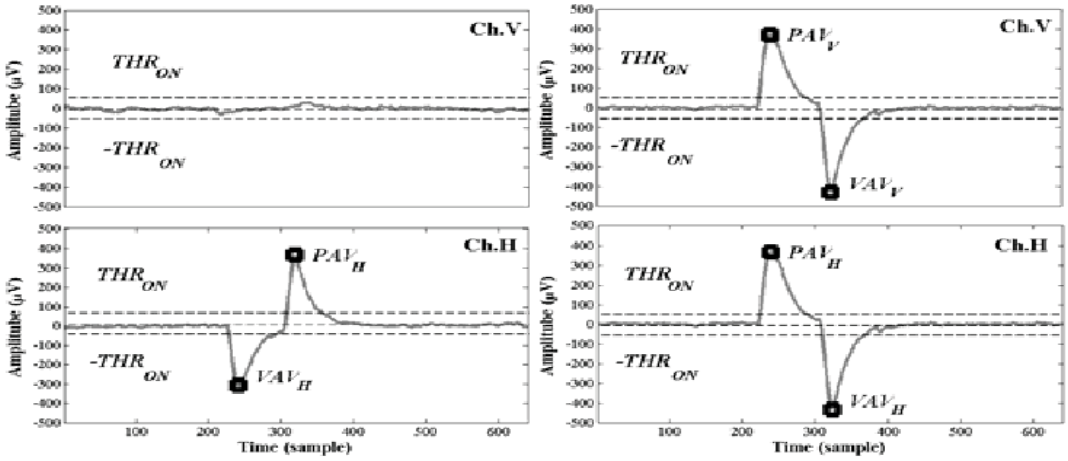

Fig. 3.4: Maximum peak and valley amplitude values

Maximum Peak Amplitude Position (PAP), Maximum Valley Amplitude Position (VAP): It's a measure of the EOG signal amplitude position at the highest point, maximum positive value of peak case and amplitude value at the lowest point, maximum negative value of valley case in both vertical and horizontal channels.

Area under the curve value (AUC): Area under the curve of the EOG signal is the summation of absolute value of the amplitude under both positive and negative curves in both vertical and horizontal channels, also upper and lower wavelengths (UWL and LWL).

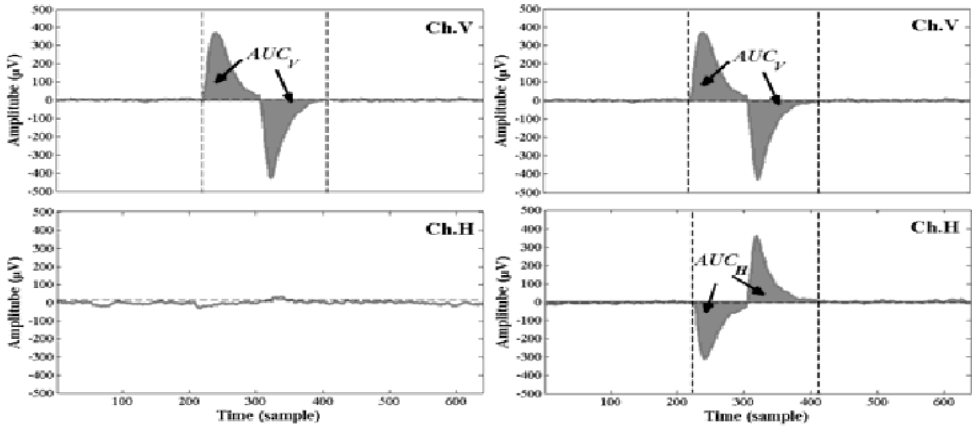

Fig. 3.5: Area under lower and upper curve

3.4 Classification:

In the classification phase, we used the feature extracted from the signal as inputs for classifiers to distinguish between the classes (Up, Down, Right, Left and Blinking), we will describe the most used in this filed, There are two types of classifiers we mention: Simple classifier: classifier depend on set of rules to determine the movements for example Thresholding, Complex classifier: such as Support Vector Machines (SVMs), K nearest neighbor (KNN), Decision tree and Convolution Neural Network (CNN) .

K-Nearest Neighbor (KNN):

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.
It assumes that similar things exist in close proximity. In other words, similar things are near to each other.

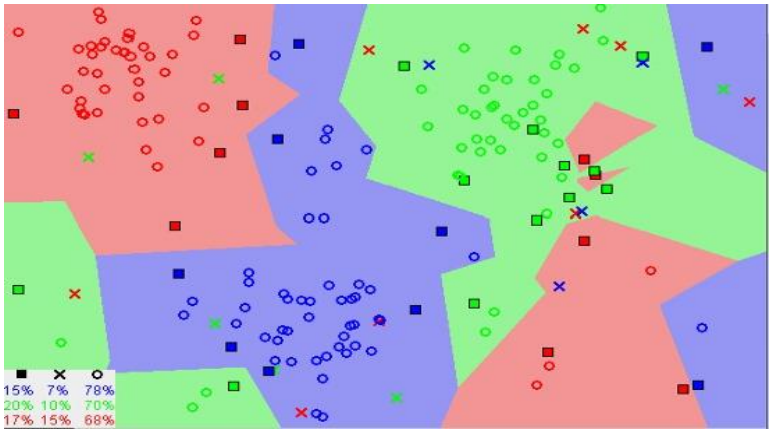

Fig. 3.6: Multi-class classification using KNN

Support Vector Machine (SVM):

Support Vector Machines are a set of supervised learning methods used for classification, regression and outliers detection.
There are many advantages for support vector machines: Effective in high dimensional spaces, Still effective in cases where number of dimensions is greater than the number of samples, Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient, Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

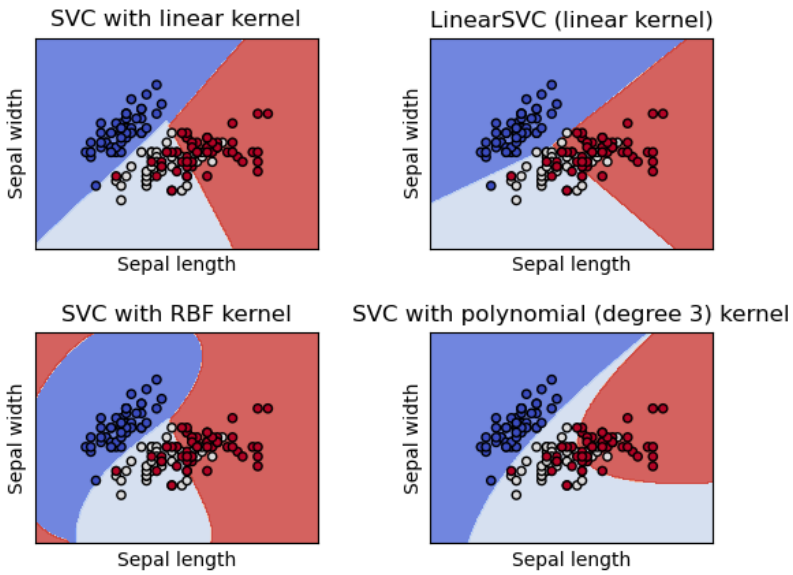

Fig. 3.7: Multi-class classification using SVMs

Decision tree:

Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.

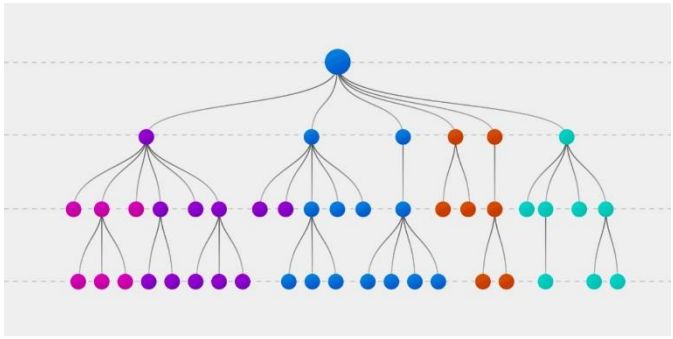

Fig. 3.8: Decision Tree

Convolution Neural Network (CNN):

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically, this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, dropout layers and activation functions.

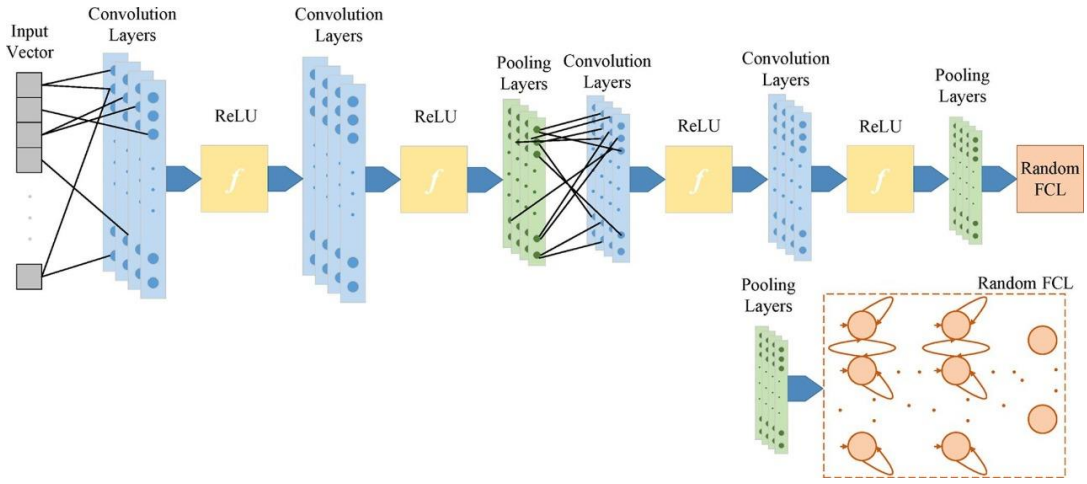

Fig. 3.9: CNN architecture

Convolution layer:

Convolution layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then we slide the filter over the next receptive field of the same input image by a Stride and do the same operation again. We will repeat the same

process again and again until we go through the whole image. The output will be the input for the next layer.
Convo layer also contains ReLU activation to make all negative value to zero.

ReLU activation function:

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

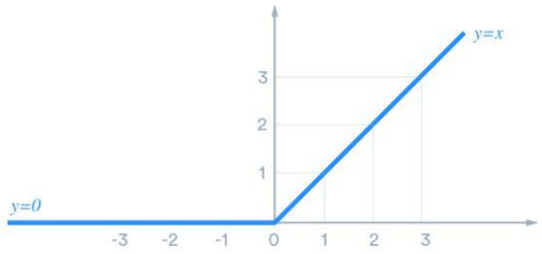

Fig. 3.10: ReLU activation function

Pooling layer:

Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If we apply FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive, and we don't want it. So, the max pooling is only way to reduce the spatial volume of input image. In the above example, we have applied max pooling in single depth slice with Stride of 2. You can observe the 4x4 dimension input is reduce to 2x2 dimension.

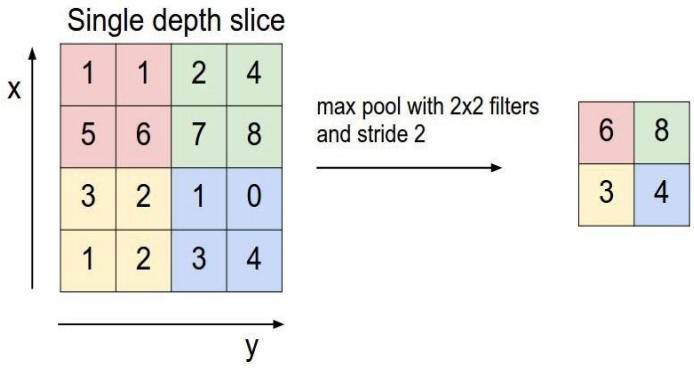

Fig. 3.11: Max pooling filter

Fully Connected Layer (FC):

It involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different category by training.

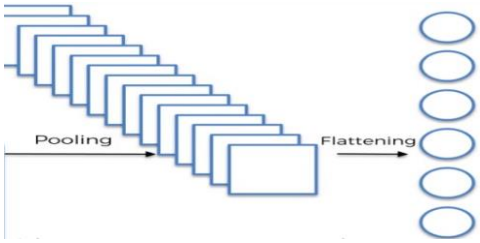

Fig. 3.12: flatten layer

Dropout layer:

Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

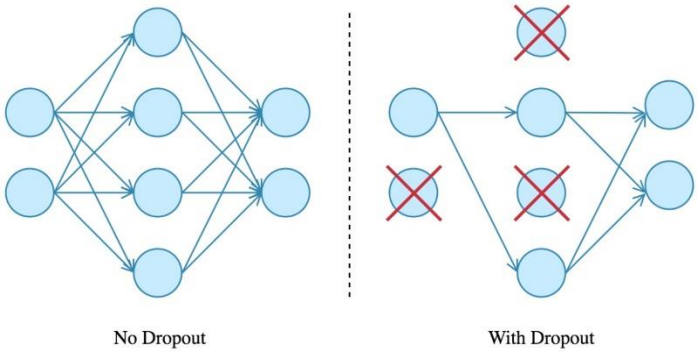

Fig. 3.13: dropout layer

Activation functions:

The activation function is a node that is put at the end of or in between Neural Networks. They help to decide if the neuron would fire or not. We have different types of activation functions.

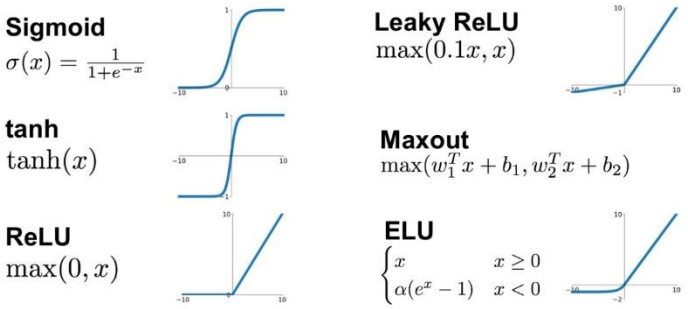

Fig. 3.14: activation function

3.5 Controlled device:

In this phase, we can take then classified eye movement and used it in the interface that can translate each classified one to action in the interface, many human computer interface applications can be done depending on eye movements: Entertainment as (playing game), robot controlling as (make a robot as helper person), wheel chair as (move the chair to walk in streets), dealing with computer as (virtual keyboard).

Fig. 3.15: controlled devices using EOG

Chapter 4:
System Implementation

4.1 Data Acquisition

4.2 Preprocessing

4.3 Classification

- **4.3.1 CNN model**
- **4.3.2 Inception V1 model**
- **4.3.3 Inception V2 model**
- **4.3.4 VGG 16 model**
- **4.3.5 ResNet model**

4.4 Reading data in real time

4.5 Windowing and Overlap

4.6 Wheelchair controlling

In this chapter, we will present the proposed system that we have implemented to develop eye-based wheelchair guidance system so that disable people can move their chair only through their eye movement, after identifying the eye movement and classifying it as one of the six movements: Up, Down, Right, Left, Blinking and No movement, we did that by applying training and testing using many deep learning classification models to determine which is the best one.

4.1 Data Acquisition:

Data acquisition (DAQ) is the process of measuring an electrical or physical phenomenon such as voltage, current, temperature, pressure, vibration or sound with a computer. A DAQ system consists of sensors, DAQ measurement hardware, and a computer with application software, this process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems, typically convert analog waveforms into digital values for processing.

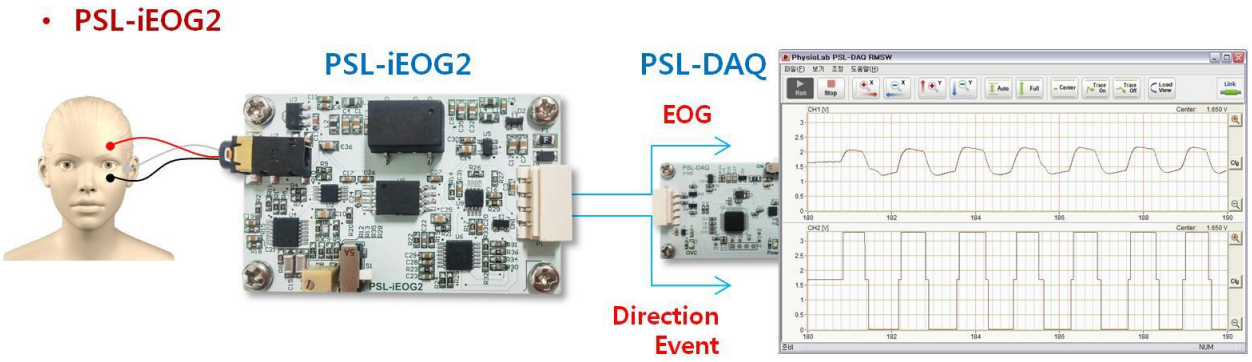

Fig. 4.1: connection of PSL-IEOG2

The components of data acquisition systems include:

- Signal conditioning circuitry: Transforms noisy signals into forms that can be effectively and accurately measured.
- Analog-to-digital converters (ADCs): Digitize analog data into digital representations that can be manipulated by computers.
- Computer bus: enables the DAQ device to transmit data to a computer. Examples include USB or Ethernet.

Software properties for our device (PSL-IEOG):

The device includes 2 parts (2-unit PSL-iEOG2) and (1-unit PSL-DAQ), Analog EOG 2-channel amplifier module (Ch.1 EOG, Ch.2 EOG Direction Event), selectable 50Hz or 60Hz by the notch filter switch , It has optimized HPF and LPF design for EOG, The output signal 0~3.3V(center 1.65V) with Sampling Rate=1,000 SPS, we can convert data format from *.pdq then to *.txt (Notepad, Excel, MATLAB compatibility)

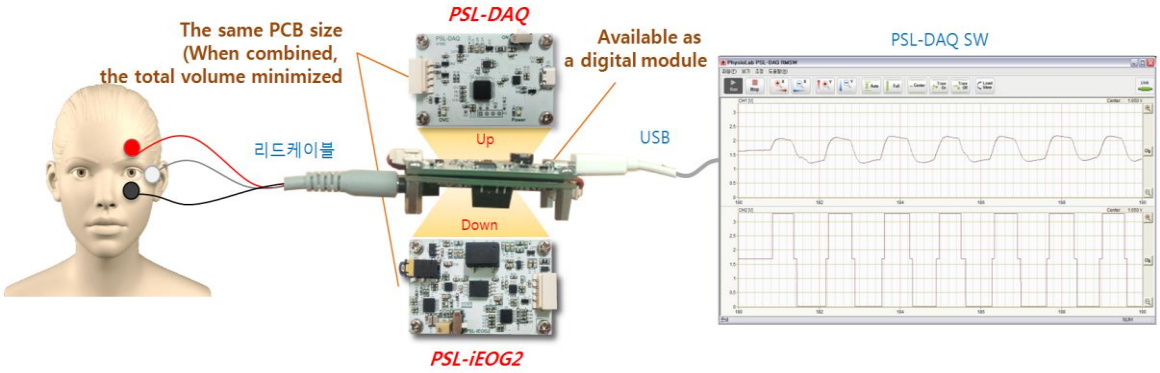

Fig. 4.2: PSL-iEOG2 and PSL-DAQ placed up-down

We have 2 units from IEOG to measure horizontal and vertical movements at the same time to achieve synchronization between them, then the output analogue signal from IEOG channel is the input to the 2nd unit (DAQ) to convert it to digital signal, finally we connect the output digital signal we connect it to pc through USB.
For Our dataset, We have applied signal measuring using the software with 50 persons every one made 10 trials (signals), each signal is separated into the previous mentioned movements through vertical and horizontal channels.so, we have 500 pairs vertically and horizontally, using 5 fold cross validation and divide data into 400 pair for training and 100 pair for testing .

4.2 Pre-processing:

It refers to the transformations applied to our data before feeding it into our models, because it directly affects the ability of our model to learn. The pre-processing techniques we had applied in two steps filtering and resampling, in filtering the acquired eye signals are contaminated by noise and artifacts. Hence, pre-processing is needed to avoid the involuntary EOG movements and keep EOG band width (0.5,20) using butter worst bandpass filter with second order.
In resampling the acquired signals don't have the same size and it causes problems like many computations cause of long signals, need to have the same dimension to

be prepared to the model .so, to solve these problems we have applied resampling for each signal to be 50 samples.
After applying pre-processing on each horizontal and vertical data file we have applied concatenation operation between them to see that the signal length is 100 depending on that : 50 resampled horizontal signal and 50 resampled vertical signals

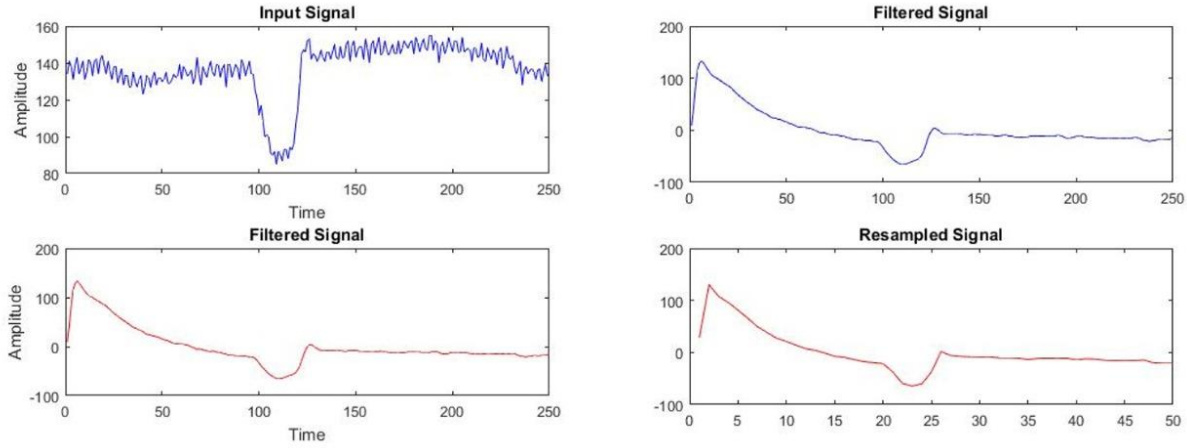

Fig. 4.3: Applying pre-processing on signal

4.3 Classification:

we applied different classifiers to choose the best one that get the best classification results.

4.3.1 1D CNN Model:

Convolutional neural network models were developed for image classification problems, where the model learns an internal representation of a two-dimensional input, in a process referred to as feature learning.
This same process can be harnessed on one-dimensional sequences of data, The model learns to extract features from sequences of observations and how to map the internal features to different eye movement.
The benefit of using CNNs for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to manually engineer input features. The model can learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features.

The proposed CNN model architecture:

It consists of 7 Convolution layer with different number of filters (32,64,512,1024) and sizes with same padding and activation function RelU With one pooling layer (Max pooling) with size 5x1 ,Then dropout layer by percentage 50% then flatten layer and dense layers with different number of filters also and activation function is ReLU and the last layer connect the last hidden layer with output layer that contain our six classes (up ,down, right, left, blinking, no movement) by their probability and classify the signal according to their maximum probability.

After building the model we have to train it we tried on many hyper parameters, but the best results were when the number of epoch 200, batch size 64 and optimizer was Adam optimizer with learning rate equal to 0.0001.

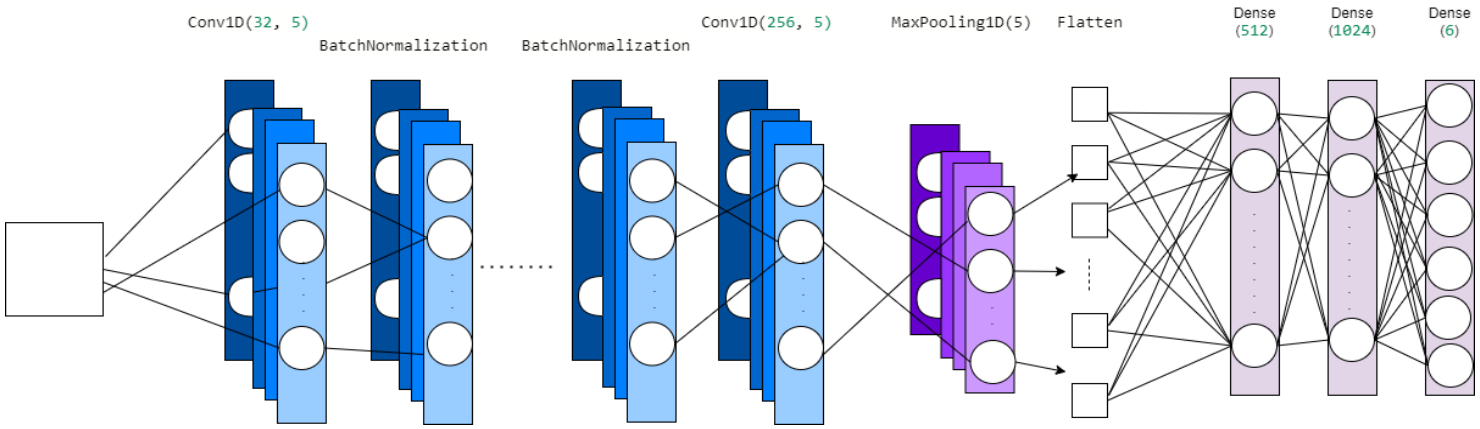

Fig. 4.4: CNN architecture

4.3.2 Inception V1 Model:

Inception V1 is based on GoogLeNet. In order to reduce the computation of 5x5 convolution, add 1x1 convolution kernel before 3x3conv, 5x5conv, and 3x3max pooling to reduce the total The role of the number of network parameters.

If the output of the previous layer is 100x100x128, after 5x5 convolutional layer with 256 outputs (stride=1, pad=2), the output data is 100x100x256. The convolution layer has a parameter of 128x5x5x256. If the output of the above layer passes through the 1x1 convolutional layer with 32 outputs (the 1x1 convolution reduces the number of channels and the feature size does not change), and then passes through the 5x5 convolutional layer with 256 outputs, the final output data is still for 100x100x256, but the convolution parameter has been reduced to 128x1x1x32 + 32x5x5x256, the number of parameters is reduced to about one-fourth of the original.

The role of the 1x1 convolution kernel:
The biggest effect of the 1x1 convolution kernel is to reduce the number of channels in the input feature map. Assuming the input is a 6x6x128 feature map, the 1x1 convolution kernel is 1x1x32 (32 channels) and the output is 6x6x32. That is, when the number of 1x1 convolution kernels is smaller than the number of channels of the input feature map, it plays a role of dimensionality reduction.

Our Inception V1 model architecture:
It consists of 2 convolution layers of different numbers of filters and size and activation function is RelU, then the inception blocks that contains convolution on an input, with 3 different sizes of filters (1x1, 1x3, 1x5). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module, we have 5 inceptionV1 module in our model the make average global pooling and the last layer connect the last hidden layer with output layer that contain our six classes (up, down, right, left, blinking, no movement) by their probability and classify the signal according to their maximum probability.
And after building the model we have to train it we tried on many hyper parameters, but the best results were when the number of epochs 100, batch size 32 and optimizer was Adam optimizer with learning rate equal to 0.0001.

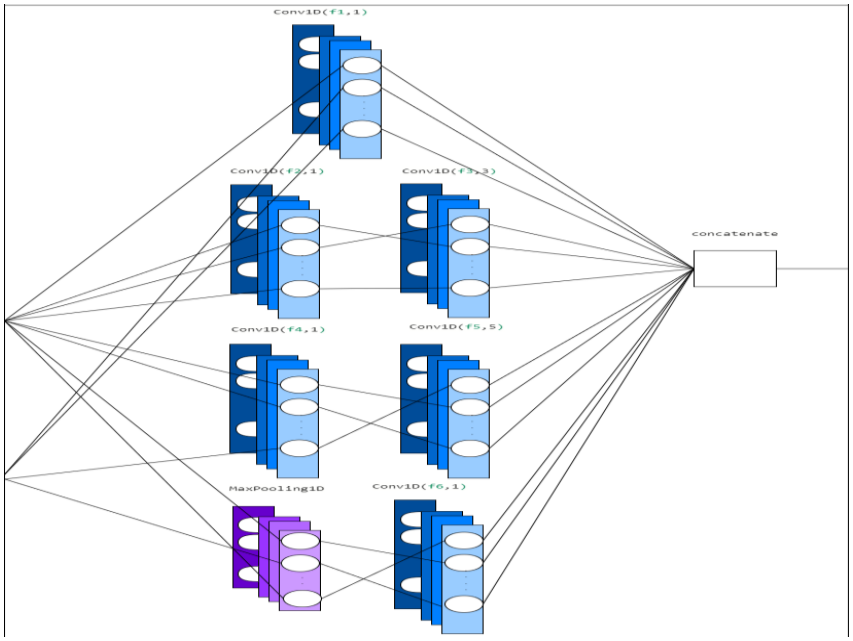

Fig. 4.5 inception V1 block

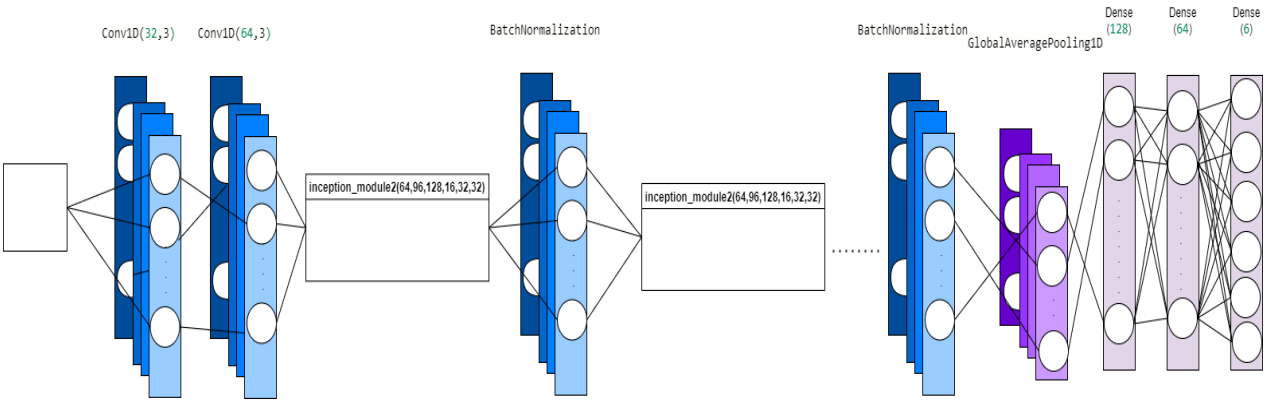

Fig. 4.6 inception V1 architecture

4.3.3 Inception V2 model:

Inception V2 has the following improvements over Inception V1: Use Batch Normalization to speed up model training; uses two 3x3 convolutions instead of a large convolution of 5x5, reducing the number of parameters and reducing overfitting;
increases the learning rate and speeds up the learning decay rate to apply BN normalized data; removes Dropout and mitigates L2 regularization (because BN has become regularized); more thoroughly disrupts training samples; Reduces optical distortion of data during data enhancement (because BN training is faster and each sample is trained less often, so more realistic samples are more helpful for training).

Our Inception V2 model architecture:

It consists of 2 convolution layers of different numbers of filters and size and activation function is RelU, then the inception blocks that contains convolution on an input, with 3 different sizes of filters (1x1, (1x1,1x3,1x3,1x1),(1x1,1x3,1x1)). Additionally, max pooling is also performed. The outputs are concatenated and sent to the next inception module, we have 5 inceptionV2 module in our model the make average global pooling and the last layer connect the last hidden layer with output layer that contain our six classes (up, down, right, left, blinking, no movement) by their probability and classify the signal according to their maximum probability.

And after building the model we have to train it we tried on many hyper parameters but the best results were when the number of epoch 100, batch size 32 and optimizer was Adam optimizer with learning rate equal to 0.0001.

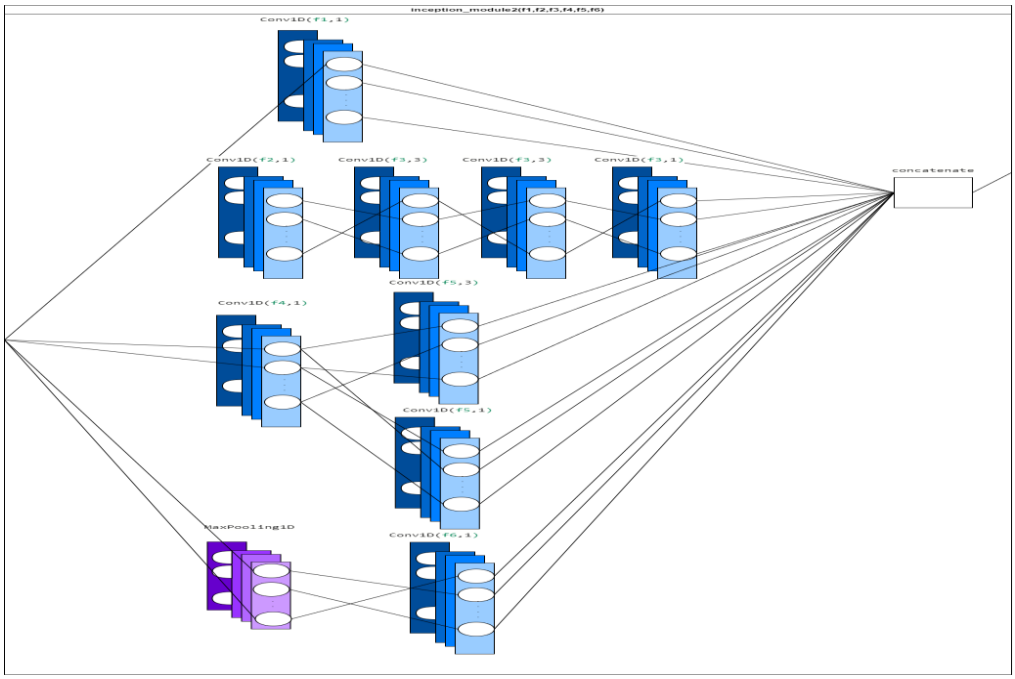

Fig. 4.7 inception V2 block

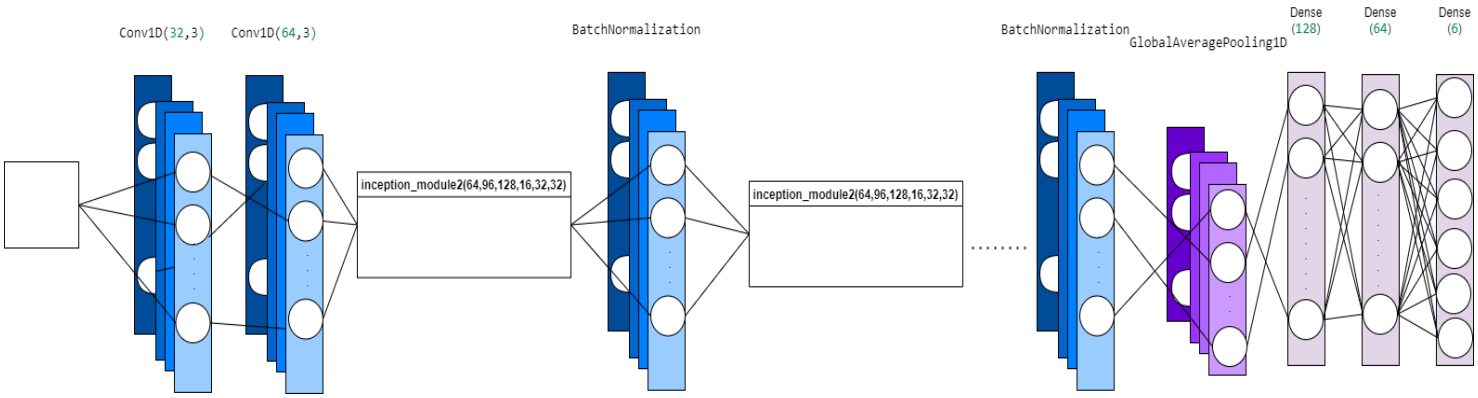

Fig. 4.8 inception V2 architecture

4.3.4 VGG 19 model:

VGG-19 is a trained Convolutional Neural Network, from Visual Geometry Group, Department of Engineering Science, University of Oxford. The number 19 stands for the number of layers with trainable weights. 16 Convolutional layers and 3 Fully Connected layers.

Our VGG 19 model architecture:

It consists of the input layer and 2 convolution layers of size (1x3) then 3 blocks of VGG19 that conation sequential layers of convolution and pooling (conv1D, conv1D, maxpooling, conv1D, conv1D, maxpooling, conv1D, conv1D, conv1D, conv1D, maxpooling, conv1D, conv1D, conv1D, conv1D, maxpooling, conv1D,

conv1D, conv1D, conv1D, maxpooling,) , Then to get the output layer we have to make a pooling layer to reduce the size of feature matrix by global maximum pooling, then the last layer connect the last hidden layers with output layer that contain our six classes (up, down, right, left, blinking, no movement) by their probability and classify the signal according to their maximum probability And after building the model we have to train it we tried on many hyper parameters but the best results was when the number of epoch 250, batch size 64 and optimizer was Adam optimizer with learning rate equal to 0.0001.

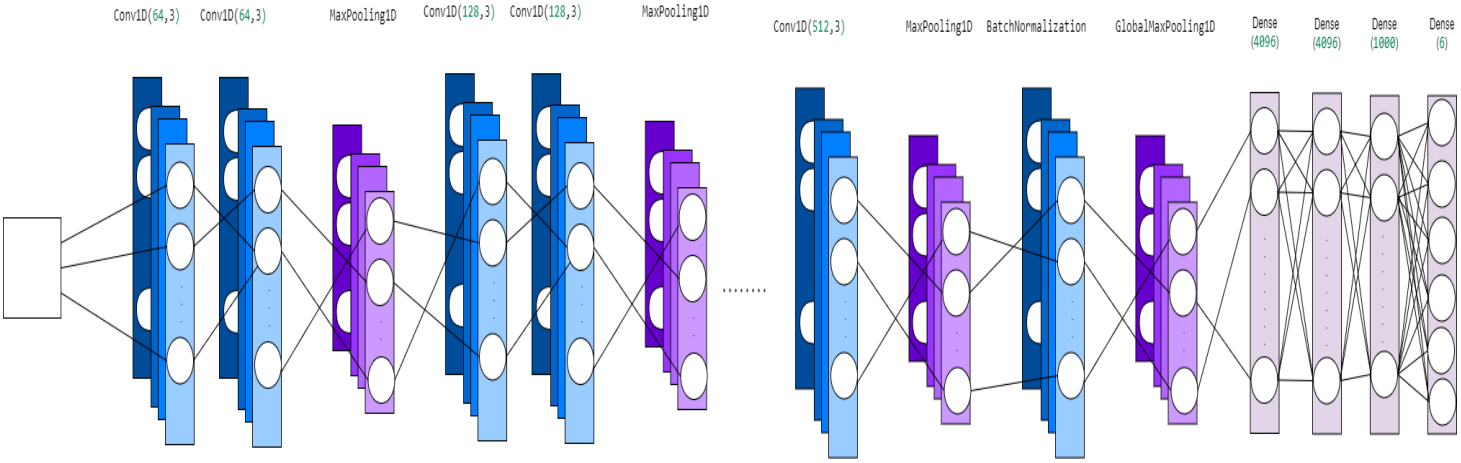

Fig. 4.9: VGG 19 architecture

4.3.5 ResNet model:

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize And can gain accuracy from considerably increased depth.

Our ResNet model architecture:

It consists of the input layer and convolution layer of size (1x7) and then repeat residual stack five times for different numbers of layers (64,128,256,512, 1024) the residual stack itself contain 3 convolution sequential layers with different filter's size and numbers then the result of them concatenated with the original input layer of the block. Then to get the output layer we have to make a pooling layer to reduce the size of feature matrix by average pooling, then flatten layer and dropout layer by percentage 50%, then the last layer connect the last hidden layers with output layer that contain our six classes (up, down, right, left, blinking, no movement) by their probability and classify the signal according to their maximum

probability And after building the model we have to train it we tried on many hyper parameters but the best results was when the number of epoch 100, batch size 32 and optimizer was Adam optimizer with learning rate equal to 0.0001.

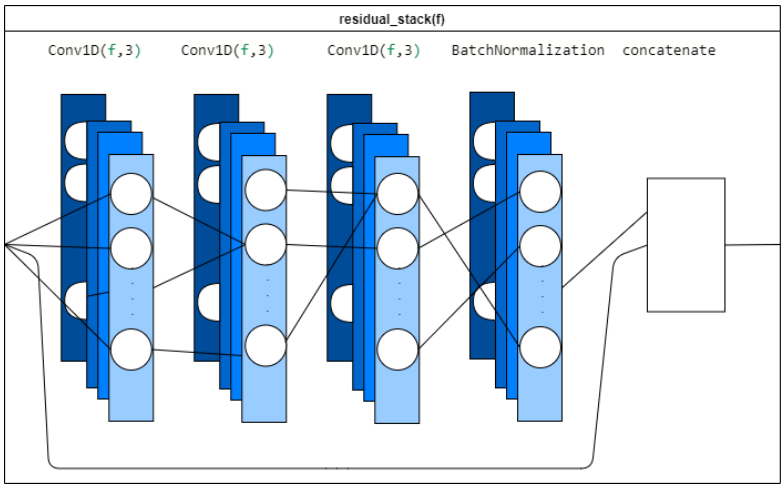

Fig. 4.10: residual block

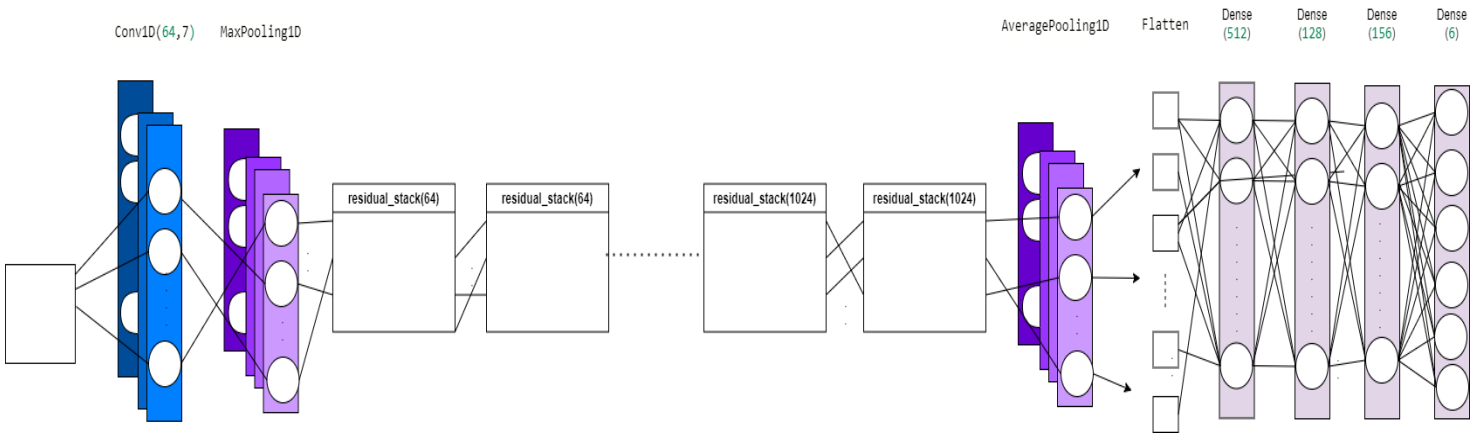

Fig. 4.11: ResNet architecture

4.4 Reading data in real time:

To get our system work in real time we had to make some changes in reading step to read continuous data and apply pre-processing and classification to get the direction for wheelchair movement.

We connect the 2-channels "vertical, horizontal" with the Arduino then connect it to the PC without using the data acquisitor chip, so that we can read the data and work on it in real time without having to save it in a file then read it again to work on it.

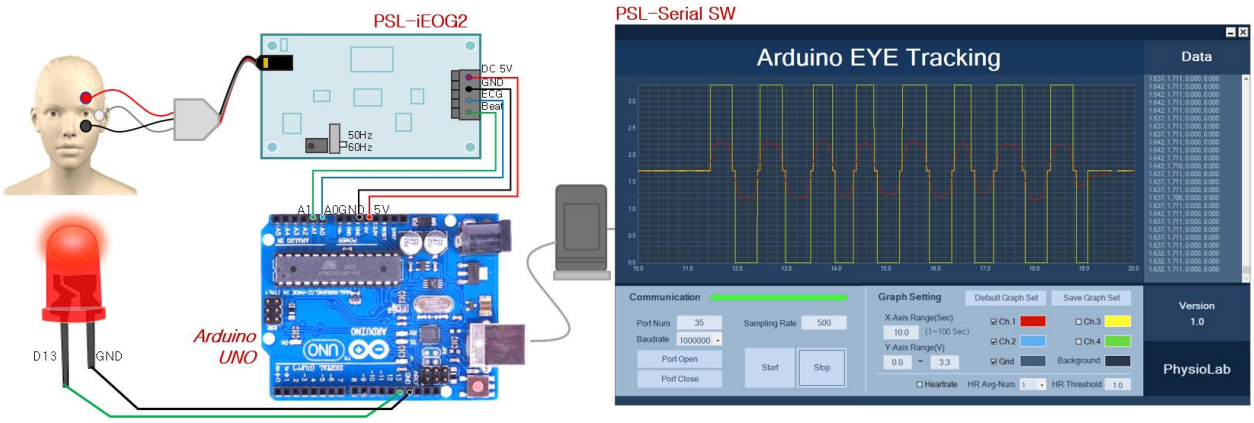

Fig. 4.12: connecting the Arduino board and PSL-iEOG2

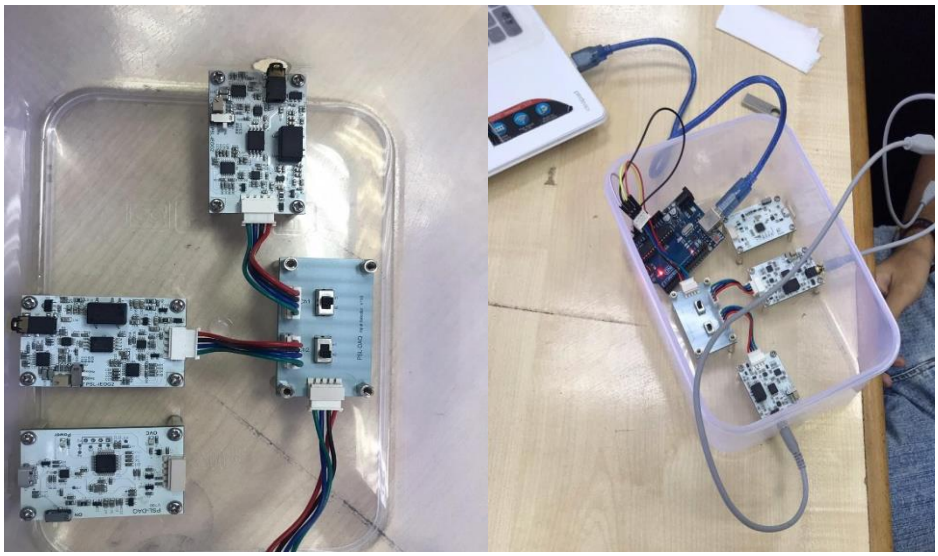

Fig. 4.13: connection between the Arduino and PSL-iEOG2

4.5 Windowing and Overlap:

the EOG signals into frames, windowing is applied to each individual frame to reduce the discontinuity at the beginning and at the end of each frame. In this the adjacent frames the length of the frames has been chosen regarding to the properties of the EOG signals. In EOG signals, the duration of blink pulses is 1900 with overlapping 240 to extract most of the features of EOG signals.

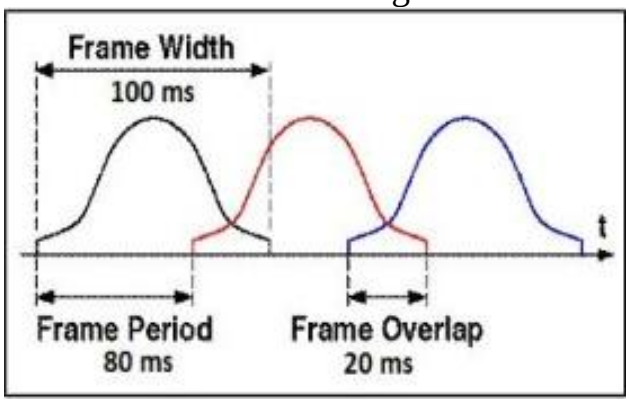

Fig. 4.14: windowing and overlapping explanation

After getting the whole signal we can apply our processes on it and classify it to get the direction.

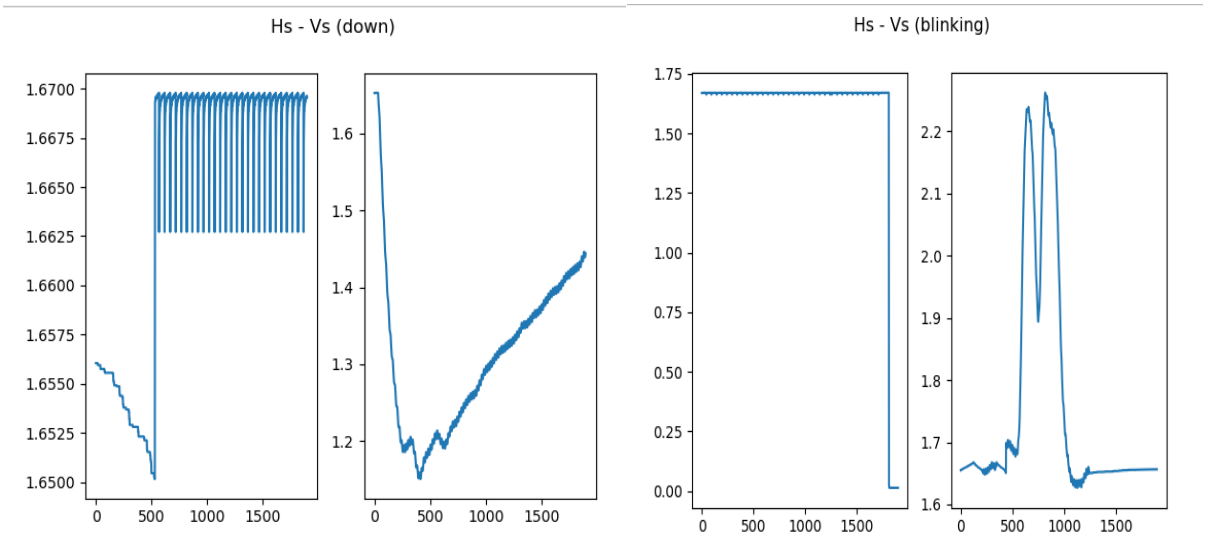

Fig. 4.15: window size and overlapping results

4.6 Wheelchair Controlling:

In this stage, after we have read the signals in Realtime, applying the windowing, and overlapping on it and classifying it to get the direction, we need to send it as an action to move the wheelchair.
We start connecting the wheelchair driver to our system using Arduino.

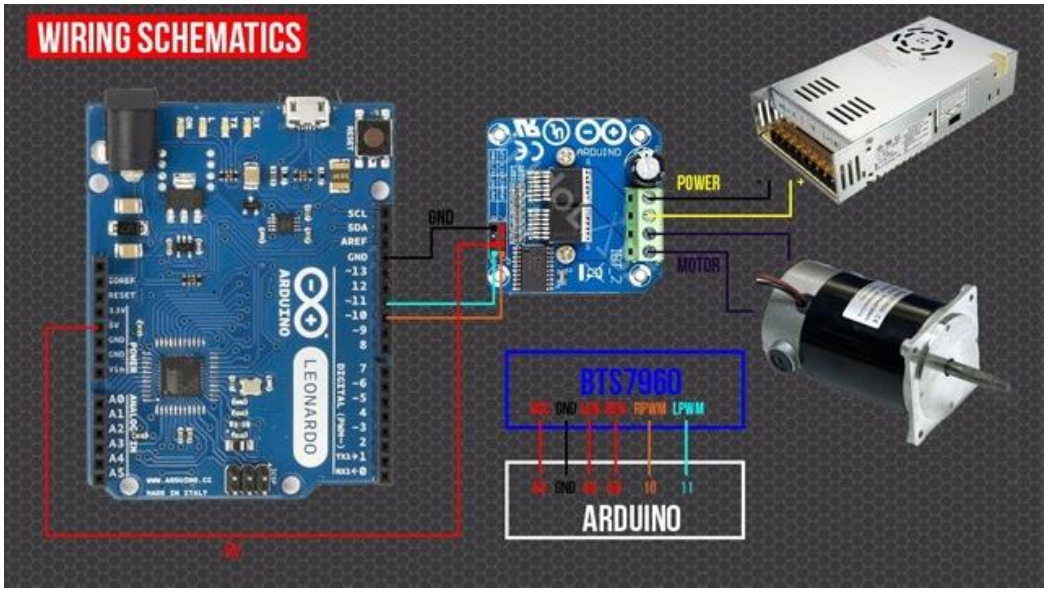

Fig. 4.16: connecting wheelchair

The connection between software and hardware:
It depends on connecting the driver and the Arduino.
Arduino is used for sending the direction order to the driver.
Driver is used for controlling the speed and direction of motors which move the wheelchair in the desired direction.

Chapter 5: System Results

5.1 Evaluation

5.2 Results

5.2.1 Results of different models

5.2.2 Best Result

5.1 Evaluation:

We evaluate our models by calculating confusion matrix that clarify right and wrong classifications in each class.
Then we use confusion matrix to calculate overall accuracy and calculate accuracy for each class then calculate average accuracy of all classes.
Overall Accuracy: number of correctly predicted direction of the signals to the actual direction.
Average Accuracy: it is the average of each accuracy per class (sum of accuracy for each class predicted/number of class)
In unbalanced datasets, overall accuracy may be not accurate to test a model.
Average accuracy is a good way to test a model to know if it has learnt all classes almost the same.
We use average accuracy over all the trials (5 trials).

5.2 Results:

5.2.1 Results of different models:

[1] CNN Model:

We try to get higher accuracy by changing the hyperparameters many times.
*First time we use #epoch = 400, batch size = 64 and #layers = 4.

Table 1 : summary result of first trial of CNN model

Trial	Accuracy
1	66.8%
2	88.8%
3	97.5%
4	85.3%
5	87.8%
Overall accuracy	85.2%

*Then we use #epoch = 1000, batch size = 64 and #layers = 4.

Table 2: summary result of second trial of CNN model

Trial	Accuracy
1	74.1%
2	92.5%
3	99.16%
4	98.16%
5	91.6%
Overall accuracy	89.3%

*Then we shuffle the data to reduce the gap between the trials and use #epoch = 300, batch size = 32 and #layers= 7.

Table 3: summary result of third trial of CNN model

Trial	Accuracy
1	94%
2	97.3%
3	96.5%
4	94.6%
5	95.1%
Overall accuracy	95.5%

*Then we use #epoch = 100, batch size = 32 and #layers= 7.

Table 4: summary result of last trial of CNN model

Trial	Accuracy
1	95%
2	96.16%
3	97%
4	95.2%
5	95.2%
Overall accuracy	95.7%

[2] Inception V1 Model:

We try to get higher accuracy by changing the hyperparameters many times.
*First time we use #epoch =100, batch size=64 and #Modules =4.

Table 5: summary result of first trial of InceptionV1 model

Trial	Accuracy
1	72%
2	89.2%
3	99.3%
4	87.2%
5	91.7%
Overall accuracy	87.9%

*Then we use #epoch = 100, batch size = 64 and #Modules = 5.

Table 6: summary result of second trial of InceptionV1 model

Trial	Accuracy
1	72.2%
2	91.5%
3	98.1%
4	88.5%
5	92.3%
Overall accuracy	88.5%

*Then we shuffle the data to reduce the gap between the trials and use #epoch = 100, batch size = 32 and #Modules = 5.

Table 7: summary result of last trial of InceptionV1 model

Trial	Accuracy
1	95%
2	96%
3	94%
4	95%
5	96%
Overall accuracy	95.2%

[3] Inception V2 Model:

We try to get higher accuracy by changing the hyperparameters many times.
*First time we use #epoch =200, batch size=64 and #Modules =2.

Table 8: summary result of first trial of InceptionV2 model

Trial	Accuracy
1	68.1%
2	81.5%
3	98.6%
4	84.8%
5	86.16%
Overall accuracy	83.8%

*Then we use #epoch = 200, batch size = 32 and #Modules = 2.

Table 9: summary result of second trial of InceptionV2 model

Trial	Accuracy
1	66.16%
2	88.8%
3	98.3%
4	85.8%
5	86.5%
Overall accuracy	85.12%

*Then we shuffle the data to reduce the gap between the trials and use #epoch = 100, batch size = 32 and #Modules = 5.

Table 10: summary result of last trial of InceptionV2 model

Trial	Accuracy
1	94%
2	86%
3	91.5%
4	95%
5	86.2%
Overall accuracy	90.5%

[4] VGG 16 Model:

We try to get higher accuracy by changing the hyperparameters many times.
*First time we use #epoch =400, batch size =32 and #Modules=3.

Table 11: summary result of first trial of VGG19 model

Trial	Accuracy
1	68%
2	82.6%
3	90.2%
4	84%
5	86.6%
Overall accuracy	82.28%

*Then we use #epoch = 250, batch size = 64 and #Modules = 3.

Table 12: summary result of second trial of VGG19 model

Trial	Accuracy
1	68%
2	90%
3	100%
4	85%
5	89%
Overall accuracy	86.3%

*Then we shuffle the data to reduce the gap between the trials and use #epoch = 250, batch size = 64 and #Modules = 3.

Table 13: summary result of last trial of VGG19 model

Trial	Accuracy
1	92.5%
2	94.8%
3	91.8%
4	94.1%
5	93%
Overall accuracy	93.2%

[5] ResNet Model:

We try to get higher accuracy by changing the hyperparameters many times.
*First time we use #epoch = 100, batch size=32 and #Modules=3.

Table 14: summary result of first trial of ResNet model

Trial	Accuracy
1	61%
2	91.1%
3	97.5%
4	87%
5	88.16%
Overall accuracy	84%

*Then we use #epoch = 100, batch size = 32 and #Modules = 5.

Table 15: summary result of second trial of ResNet model

Trial	Accuracy
1	71%
2	96.5%
3	98%
4	90.6%
5	92%
Overall accuracy	89.8%

*Then we shuffle the data to reduce the gap between the trials and use #epoch = 100, batch size = 32 and #Modules = 5.

Table 16: summary result of last trial of ResNet model

Trial	Accuracy
1	93.5%
2	95.3%
3	96.1%
4	94.8%
5	94.3%
Overall accuracy	94.8%

5.2.2 Best models:

3rd Best: is the ResNet model after shuffling the data and choose the best hyperparameters.

Table 17: best results of ResNet model

Trial	Blink	Up	Down	Right	Left	No Mov.	Accuracy
1	97	89	94	98	95	88	**93.5%**
2	99	92	94	94	99	94	**95.3%**
3	96	95	97	97	98	94	**96.1%**
4	97	96	97	94	92	93	**94.8%**
5	98	94	96	95	89	94	**94.3%**
Overall accuracy	97.4	93.2	95.6	95.6	94.6	92.6	**94.8%**

2nd Best: is the Inception V1 model after shuffling the data and choose the best hyperparameters.

Table 18: best results of Inception V1 model

Trial	Blink	Up	Down	Right	Left	No Mov.	Accuracy
1	95	94	92	100	95	91	**95%**
2	97	94	97	95	98	97	**96%**
3	96	93	92	98	97	86	**94%**
4	97	92	95	96	97	95	**95%**
5	98	95	94	97	97	95	**96%**
Overall accuracy	96.6	93.6	94	97.2	96.8	92.8	**95.2%**

1st Best: is the CNN model after shuffling the data and choose the best hyperparameters.

Table 19: best results of CNN model

Trial	Blink	Up	Down	Right	Left	No Mov.	Accuracy
1	95	95	98	99	98	85	**95%**
2	98	92	98	96	96	97	**96.17%**
3	96	94	97	99	99	97	**97%**
4	96	95	99	94	93	94	**95.17%**
5	97	92	97	96	97	92	**95.17%**
Overall accuracy	96.75	93.25	97.75	96.25	96.25	95	**95.88%**

Chapter 6: User Interface

6.1 GUI

6.1.1 Offline Mode

6.1.2 Real-time Mode

6.2 Wheelchair

In this chapter, we will present the interface of the system that we have implemented all of it depending only on the eye movement.

6.1 GUI: We create 2 modes of graphical user interface.

6.1.1 Offline Mode: it works on the data from the file which we have already read and save.

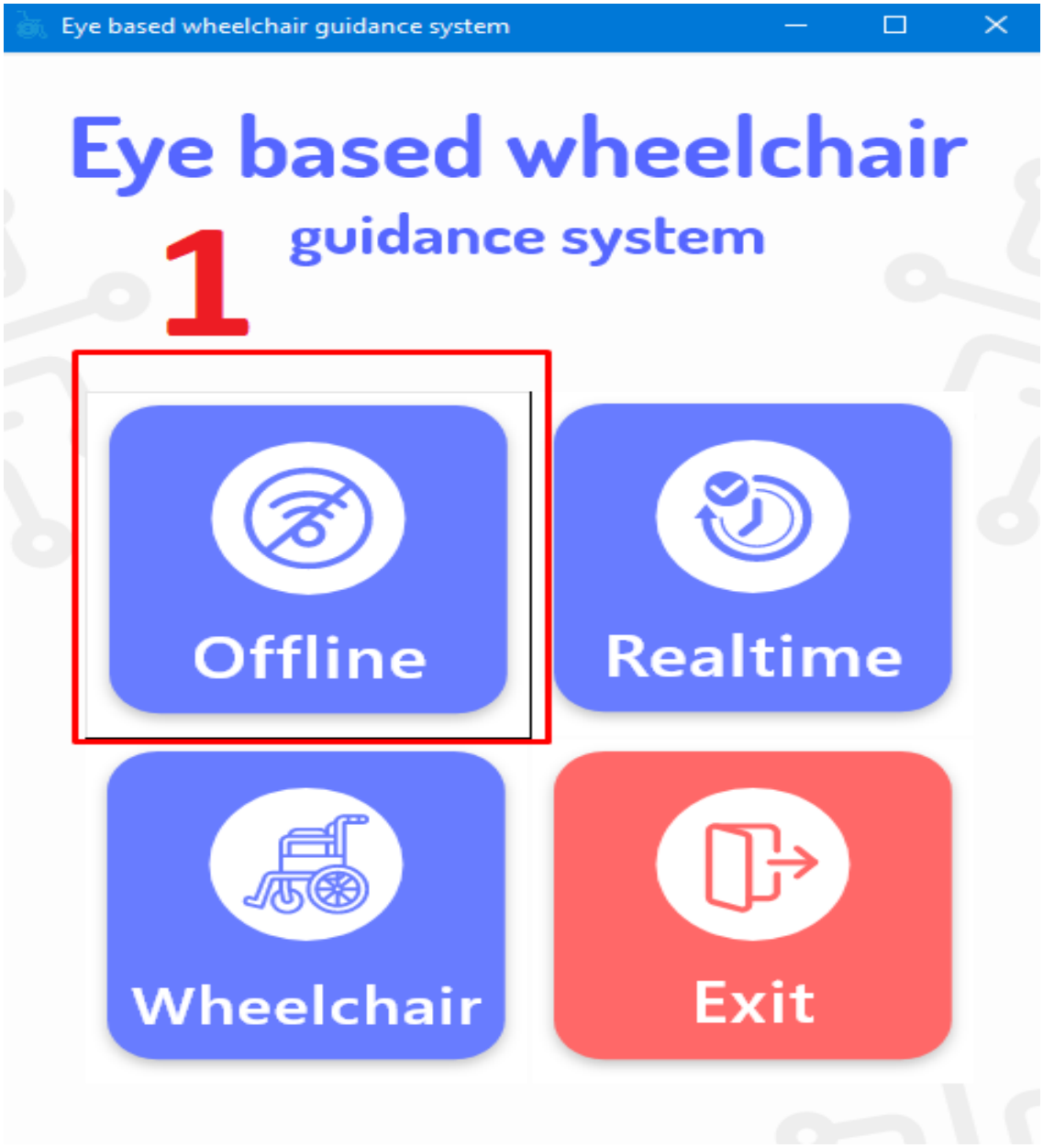

Fig. 6.1: Offline mode

By clicking on this button, it will open the next window.

1. It can import any signal file from saved files.

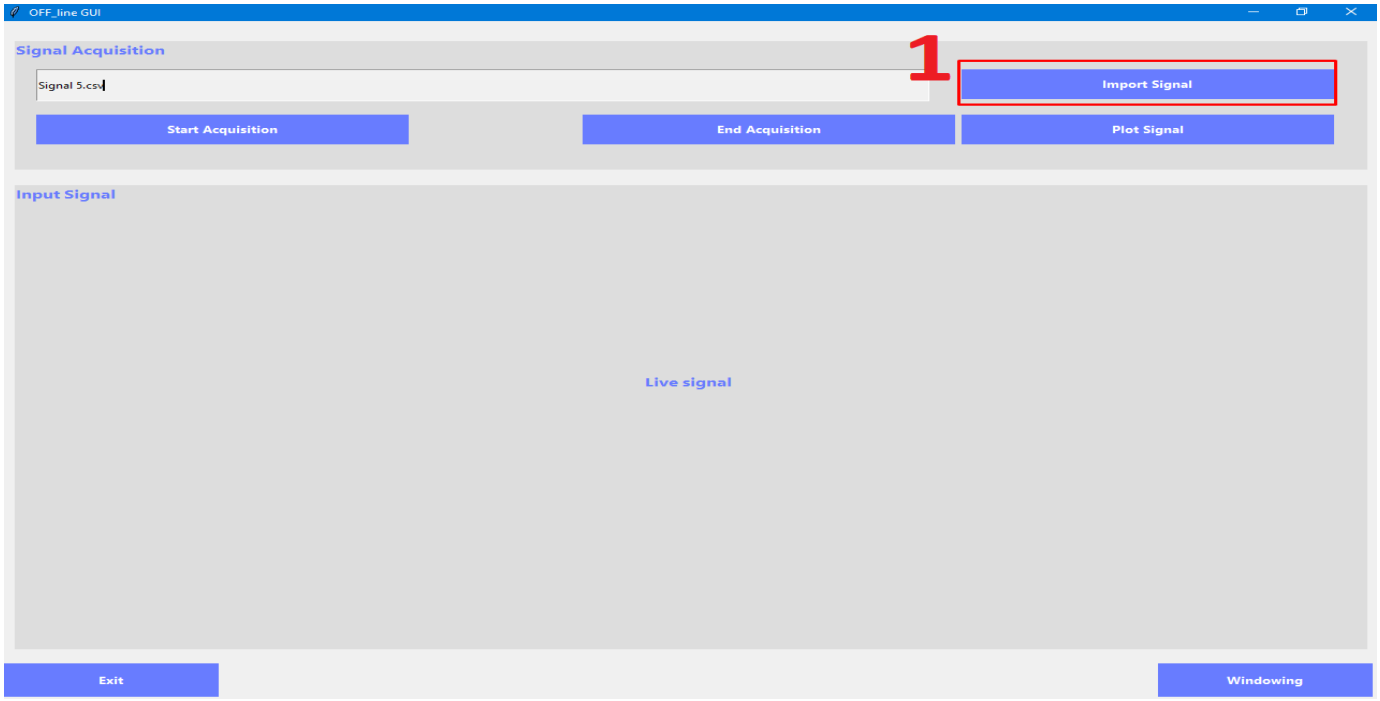

Fig. 6.2: import signal

2. It will plot the signal which have been imported.

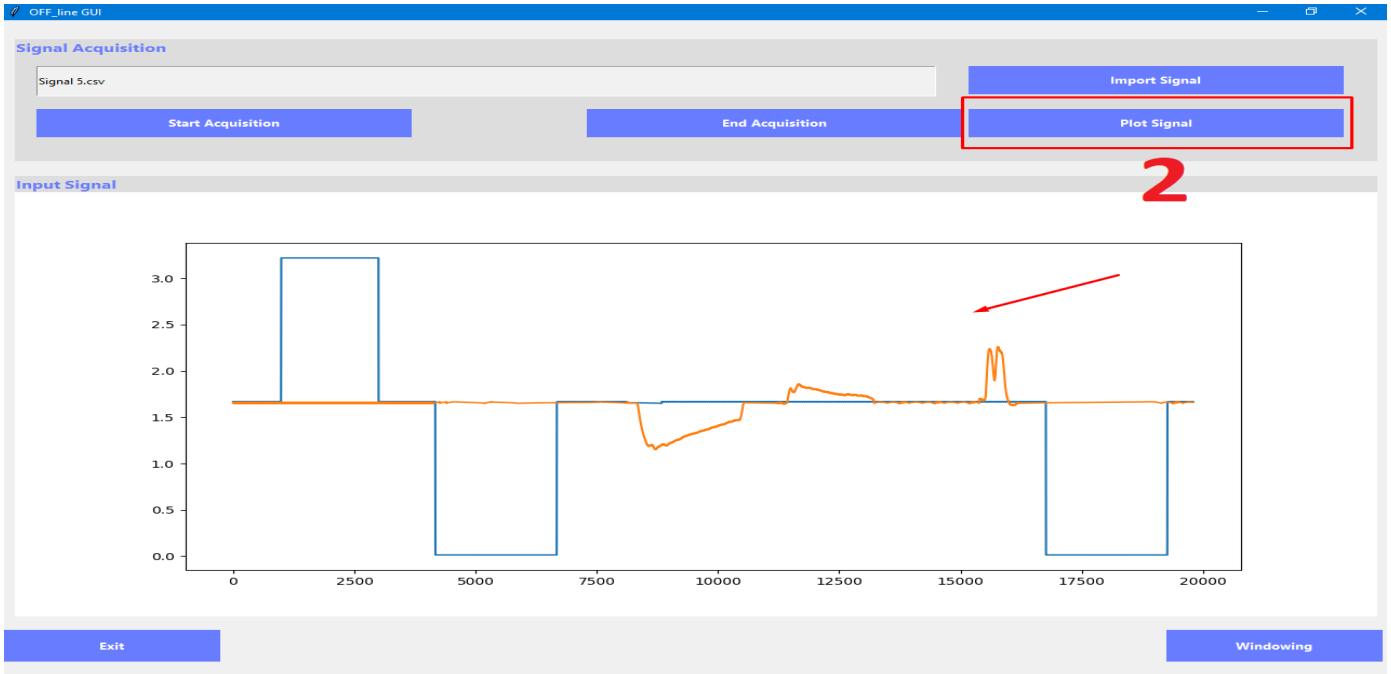

Fig. 6.3: plot signal

3. It applies windowing on the signal.

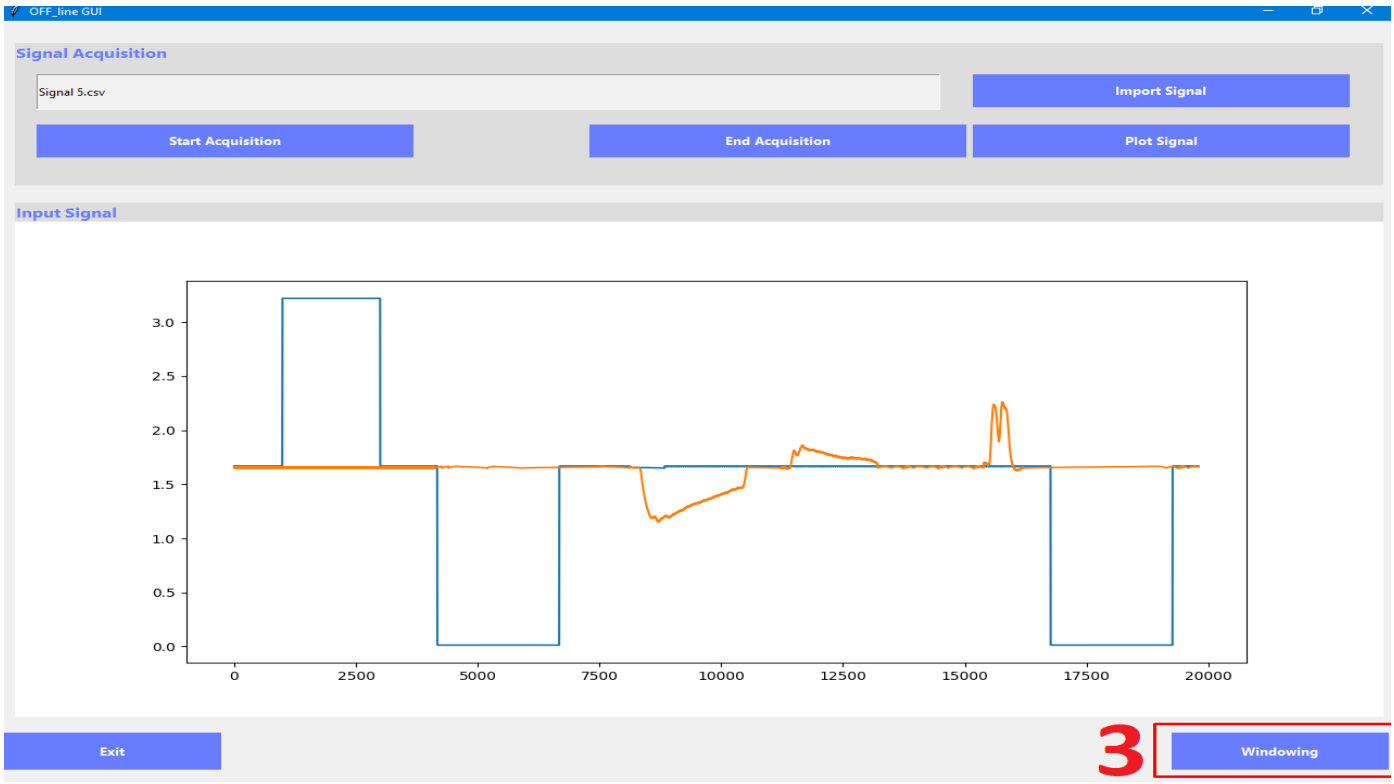

Fig. 6.4: apply windowing

4. User can enter different window size and overlapping or leave it with default values (1900, 240).

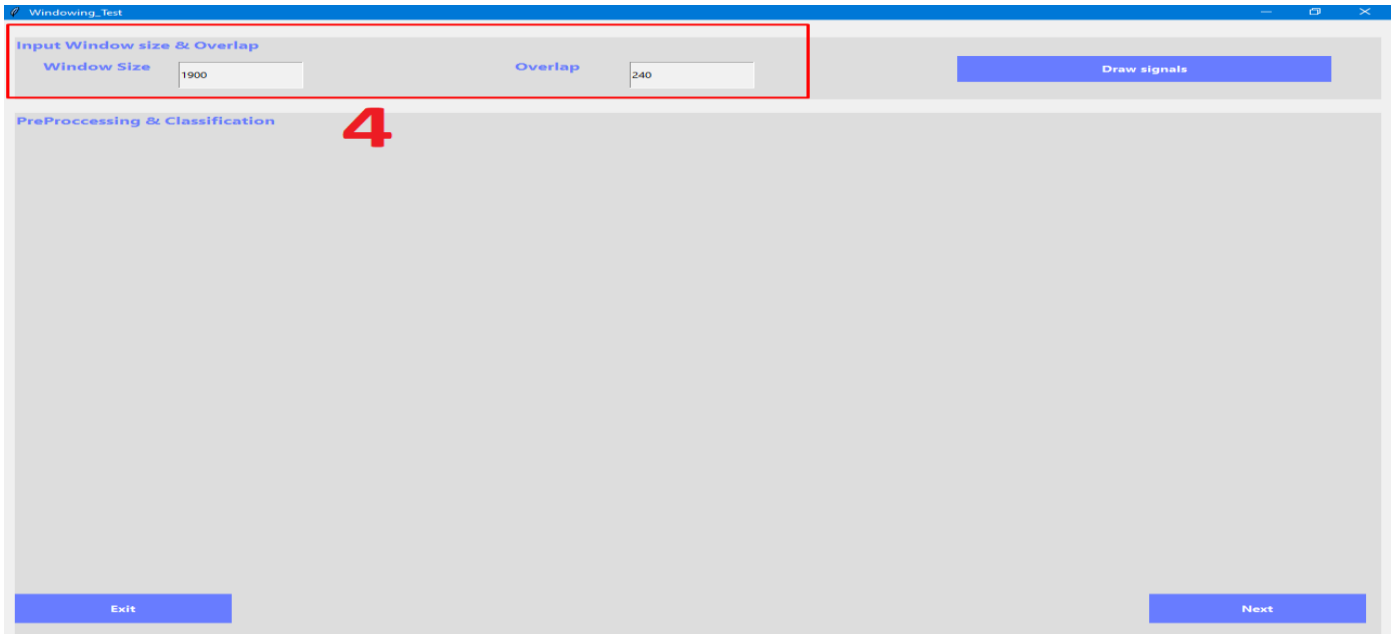

Fig. 6.5: choose window size and overlapping

5. It will start windowing with input size.

Fig. 6.6: draw signal

6. The original signal which it will use.

Fig. 6.7: original signal

7. the previous window will be updated each time.

Fig. 6.8: previous window

8. current window of the signal.

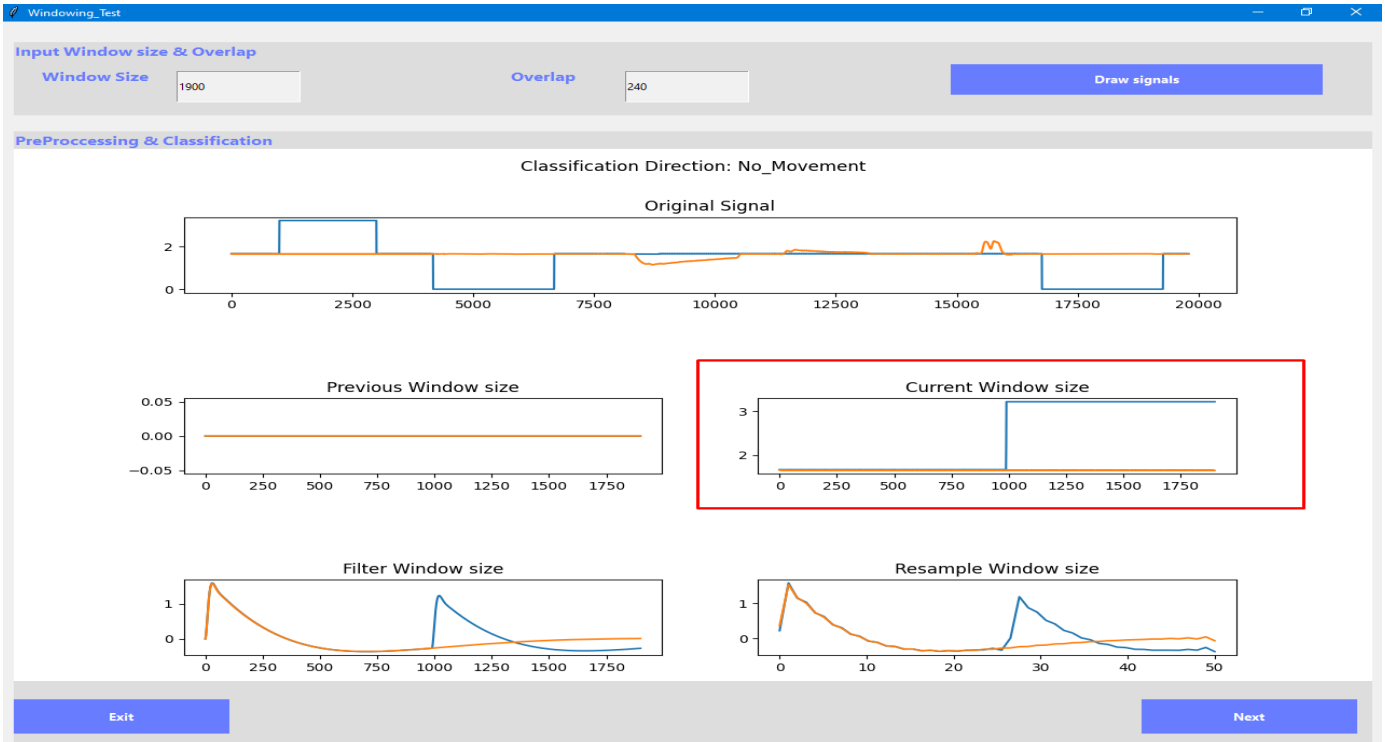

Fig. 6.9: current window

9. It take the window and apply our filter on it.

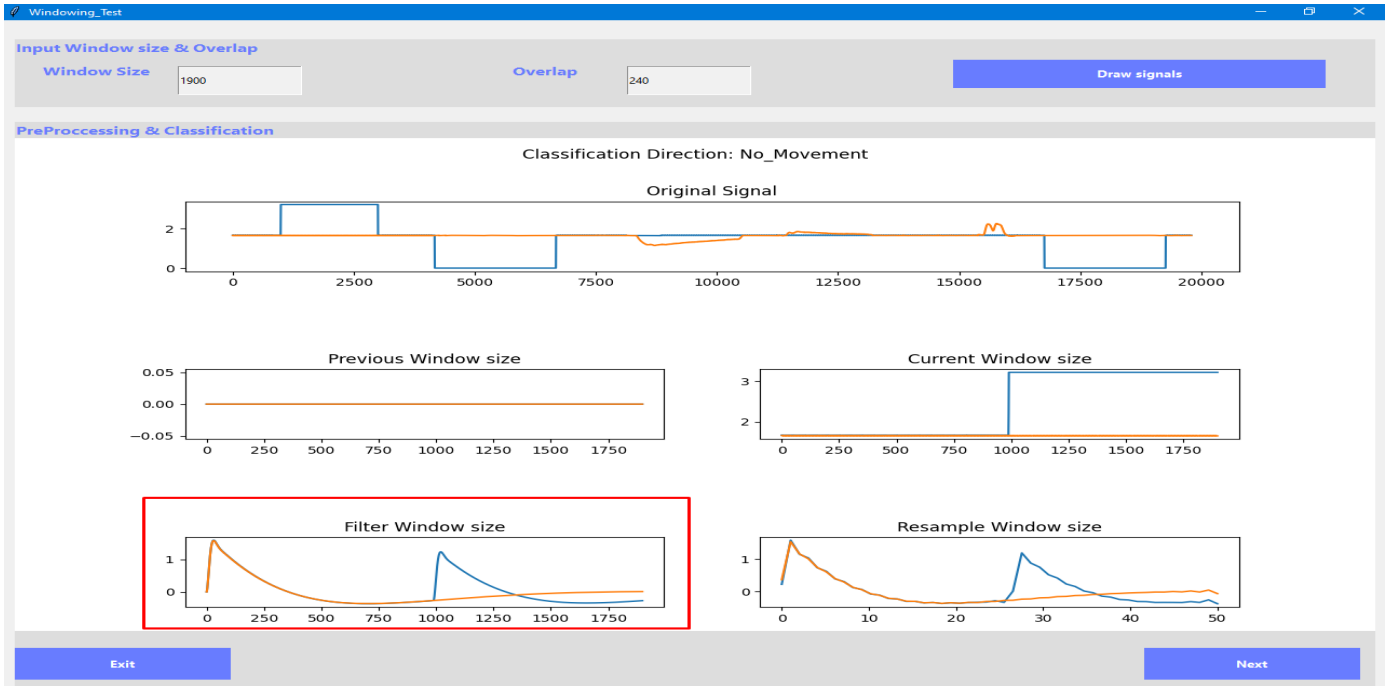

Fig. 6.10: filtered current window

10. It also apply resample to 50 for each direction.

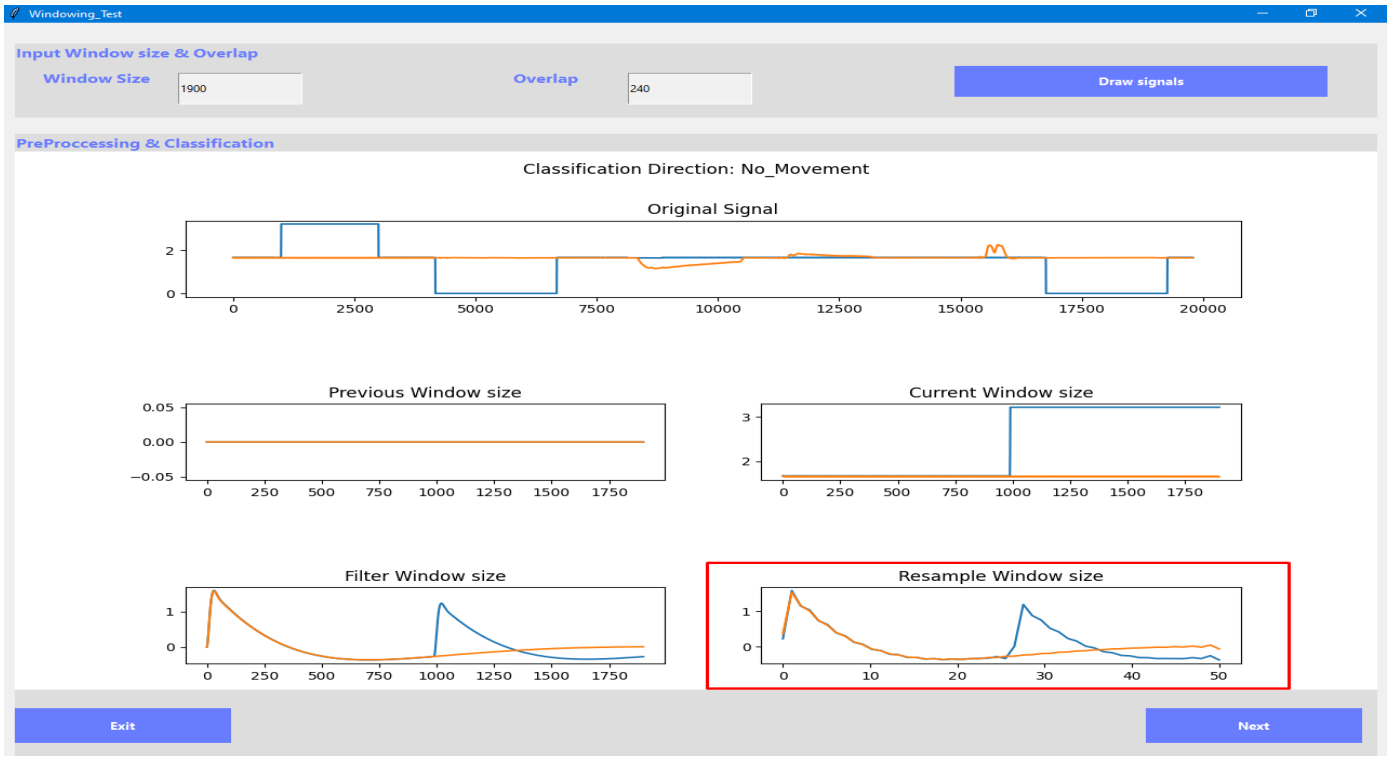

Fig. 6.11: resampled current window

11. Then apply classification on it and get the direction.

Fig. 6.12: predicted direction

12. It gets the next window from the signal, repeat all the previous steps till it get the last window of the signal.

Fig. 6.13: next window

The second option we have here is to do the same steps but on real time signal not a saved file.

By clicking on start acquisition button the system will start reading signals till end acquisition button is clicked.

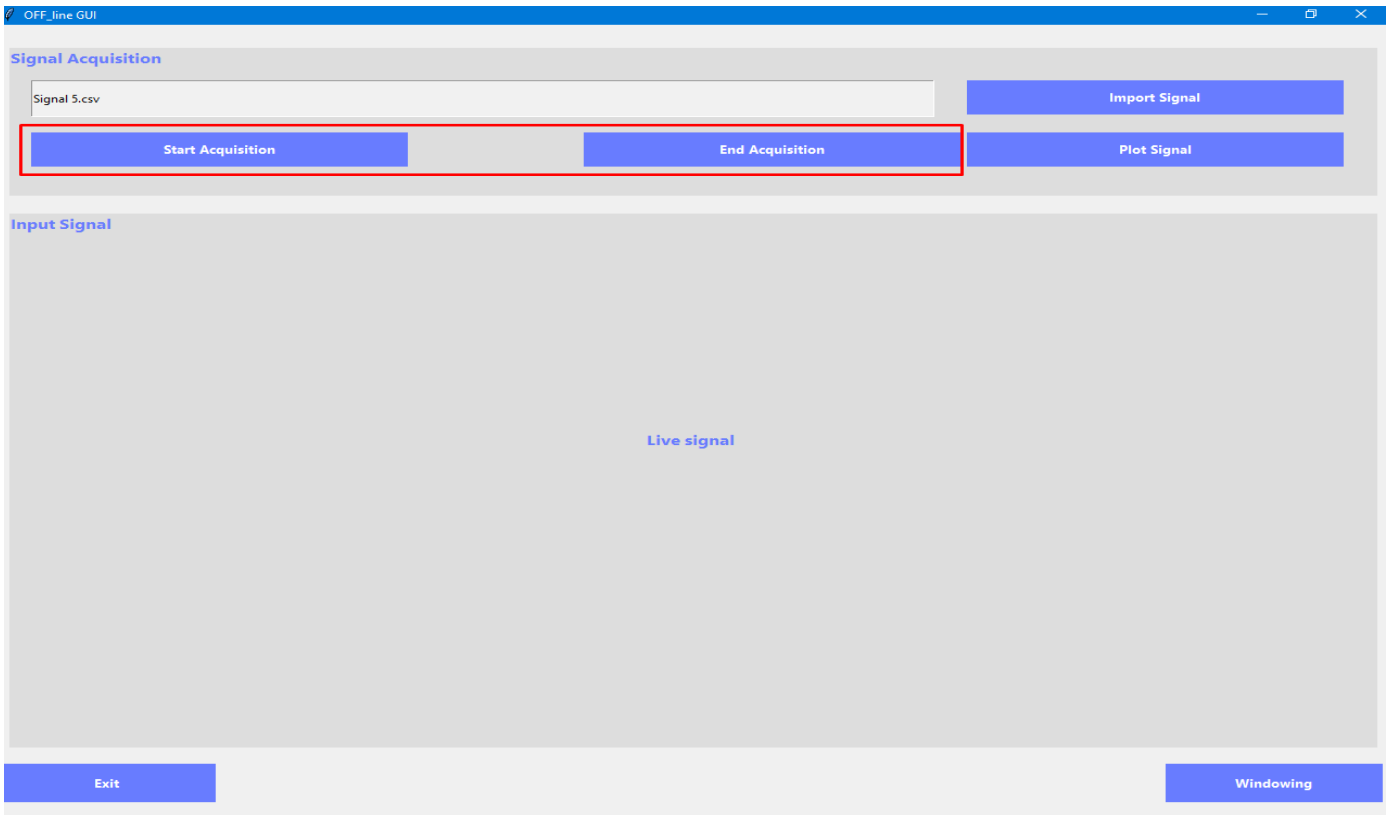

Fig. 6.14: start and end acquisition

6.1.2 Realtime Mode: it works on the real time data while reading it from the port.

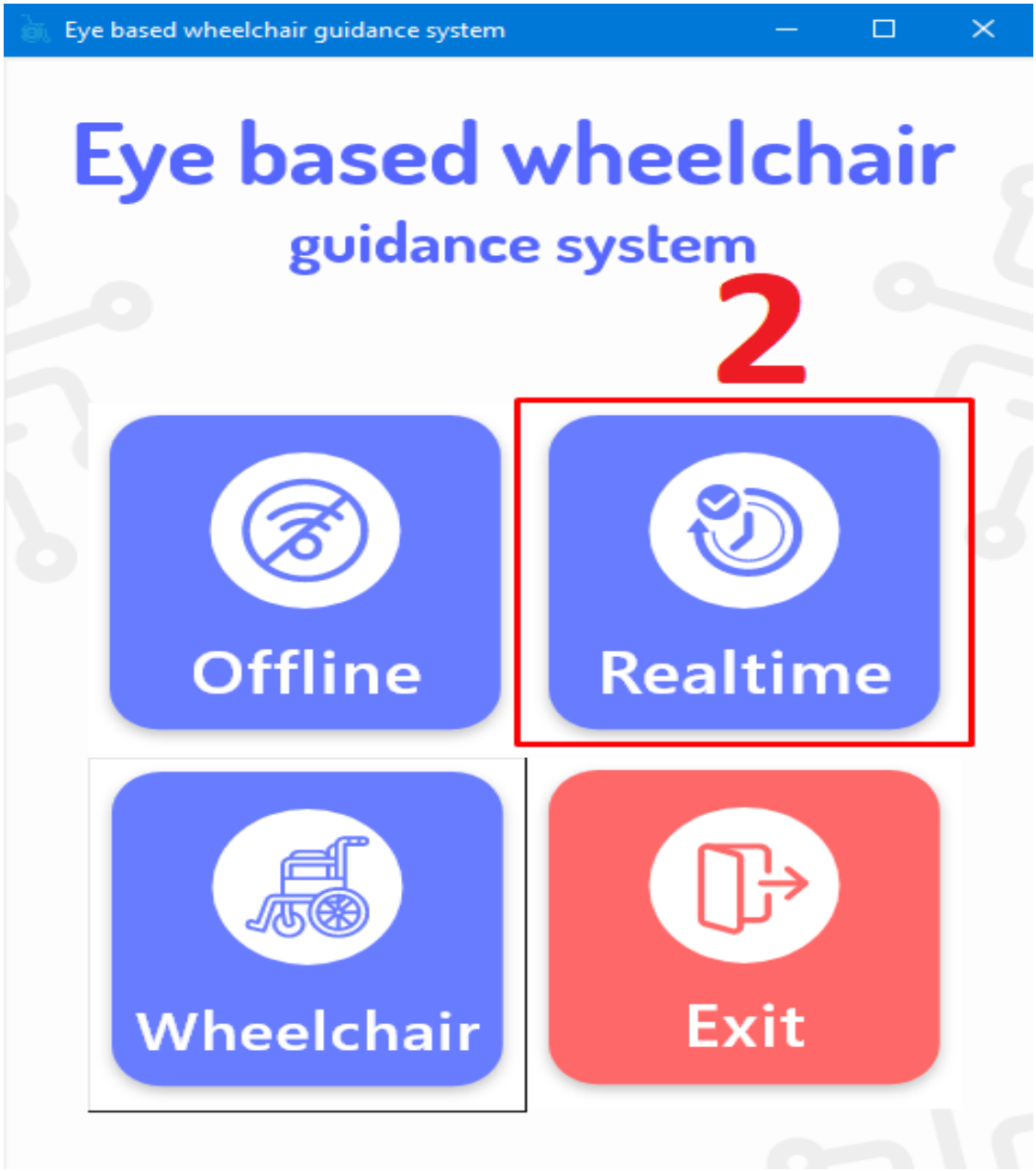

Fig. 6.15: real time mode

we have a simple game…

Fig. 6.16: real time game

It gets the direction from the model which classify each window of the input signal in real time and move according to it.

6.2 Wheelchair: when the user chooses to control his wheelchair, he only makes a double blink by his eyes to start move due to his eye direction.

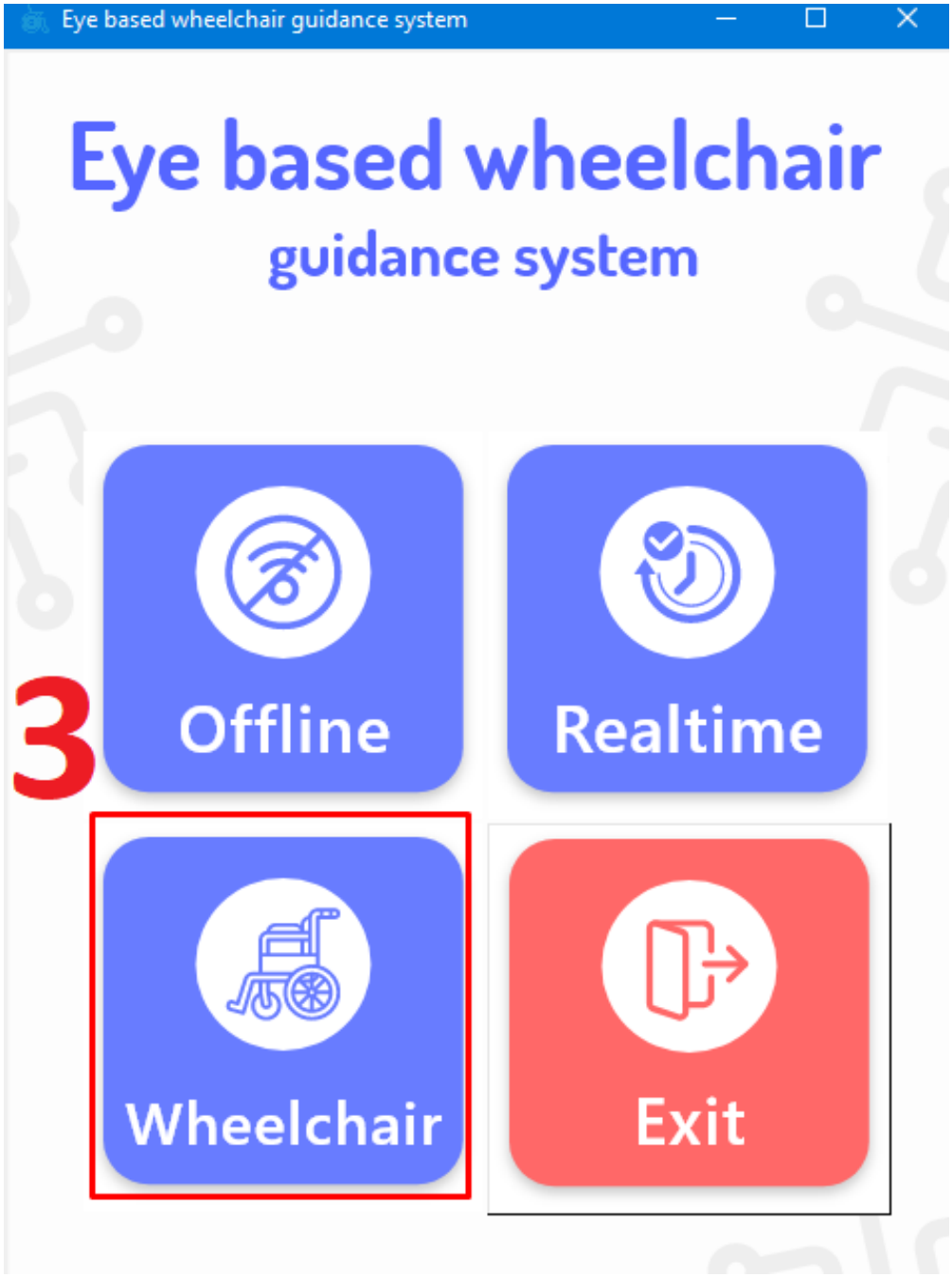

Fig. 6.17: start wheelchair moving

And here is the exit button to close the system.

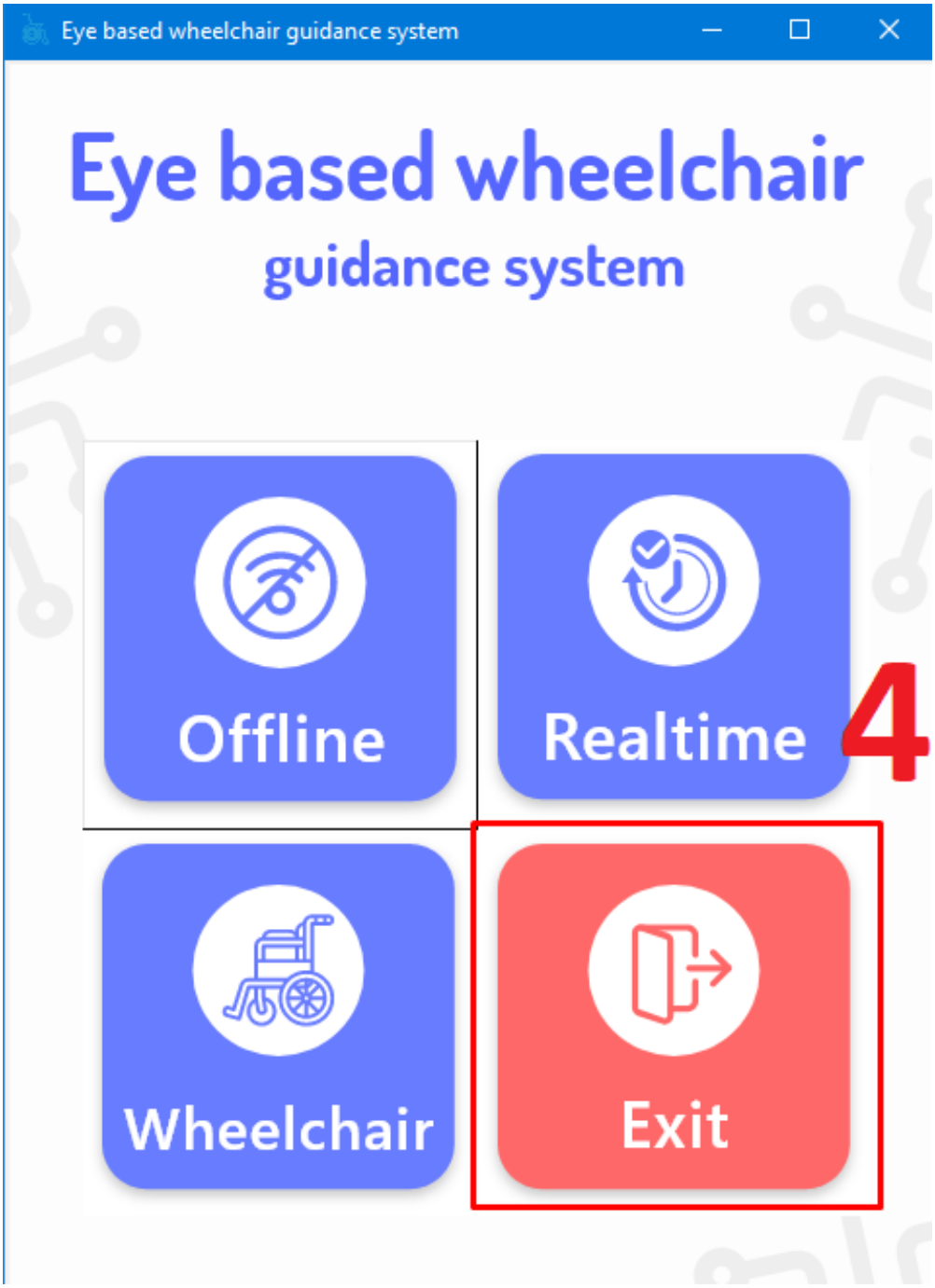

Fig. 6.18: exit

Chapter 7:

Conclusion & Future Work

7.1 Conclusion

7.2 Future Work

7.1 Conclusion:

In this work, a wheelchair based on EOG technology is presented, EOG signal reads Realtime from an EOG device by Arduino then passing it in preprocessing stage to reduce the noise by band pass filter [0.5,20], and resampled to 50 for each channel vertical and horizontal then to classification stage that can be done by more than one model (CNN, Inception_V1, Inception_V2, VGG19, Resnet). And get the best accuracy by CNN which was [95.8%] to classify the signal to six classes (Up, Down, Right, Left, Double-Blink, and No-Movement), then by Arduino and driver take the classification result direction to make by it order to move the wheelchair.

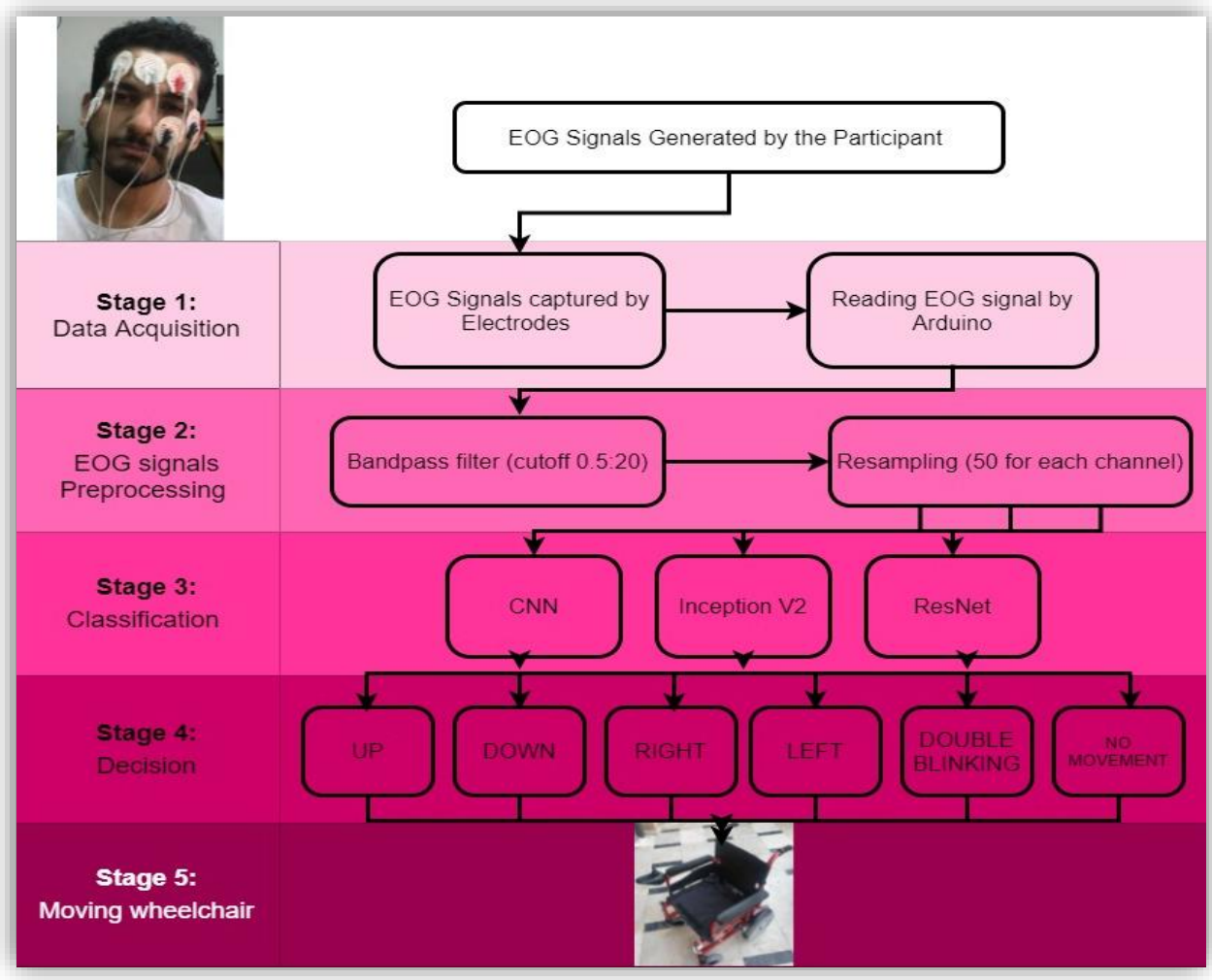

Fig. 7.1: Summary of the system

7.2 Future Work:

After what was mentioned above about our development of the eye-based wheelchair guidance system and operating it in real time , this is not the end, but the beginning of further research and development, and this is not only a vision, but specific and studied steps by us, that we will work in the future to make it more usable and save by adding some features such as: texting someone and playing a game, and we also want to improve the system to reach higher efficiency and accuracy than the investigated, as well as work to reduce the calculations and processing time.

Tools:

- Python
- Arduino
- PSL-EOG2
- Wheelchair
- Google Colab

References:

[1] Yamagishi. Development of EOG-Based Communication System Controlled by Eight-Directional Eye Movements, Annual International Conference of IEEE 2006.

[2] Jeong Heo, Heenam Yoon, Kwang suk park. A Novel Wearable forehead EOG Measurement System for Human Computer Interfaces, sensor, (2017).

[3] M. Merino, O. Rivera, I. Gomez, Al. Molina and E. Dorrenzoro, "A Method of EOG Signal Processing to Detect the Direction of Eye Movements", 2010 First Int. Conf. of Sensor Device Technologies and Applications, pp: 100-105, Italy, Jul. 18-25(2010).

[4] Rui Zhang. An EOG-Based Human-Machine Interface to Control a Smart Home Environment for Patients With Severe Spinal Cord Injuries, IEEE Transactions on Biomedical Engineering, 2018.

[5] Kwang-Ryeol. Real time "eye writing" recognition using Electrooculogram(EOG), IEEE 2016.

[6] Adarsh Rajesh, and Megha Mantur, Eyeball Gesture Controlled Automatic Wheelchair Using Deep Learning,Conference Paper · December 2017.

[7] Hema.C.R, Paulraj.M.P & Ramkumar.S. Classification of Eye Movements Using Electrooculography and Neural Networks. IJHCI, (51) 2014.

[8] Dr. S. Ramkuma, G. Emayavaramban. EOG signals classification using neural network human computer interaction. IJCTA, 9(24), pp. 223-231, 2016.

[9] B. Estrany. EOG signal processing and analysis for controlling computer by eye movements, 2009.

[10] Anwesha Banerjee,Classifying Electrooculogram to Detect Directional Eye Movements,Modeling Techniques and Applications (CIMTA) 2013

[11] Majaranta, P.; Räihä, K.-J. Twenty years of eye typing: Systems and design issues. In Proceedings of the (2002).

[12] Webster, J. Medical Instrumentation: Application and Design; John Wiley & Sons: Hoboken, NJ, USA (2009).

[13] Zhao Lv, Xiao-Pei Wu1, Mi Li, De-Xiang Zhang, Development of a human computer Interface system using EOG. Ministry of Education China, of Anhui University, Hefei, Anhui (2009).

[14] T. O. Ya, M. K. Asumi, Development of an input operation for the amyotrophic lateral sclerosis communication tool utilizing EOG, Medical and Biological Engineering, 43(1), 172-178, In Japanese (2005).
[15] Rowland, L.P.; Shneider, N.A. Amyotrophic lateral sclerosis. N. Engl. J. Med. (2001).
[16] Majaranta, P.; Räihä, K.-J. Twenty years of eye typing: Systems and design issues. In Proceedings of the Symposium on Eye Tracking Research & Applications, New Orleans, LA, USA, (2002).
[17] Nathan D S, Vinod A P and Thomas P K, "An Electrooculogram based Assistive Communication System with Improved Speed and Accuracy Using Multi-Directional Eye Movements", 35th International Conference on Telecommunications and Signal Processing (TSP), pp. 554-558, Prague, Czech Republic (2012).
[18] L. Y. Deng, C.L. Hsu, T.Ch. Lin, J.S. Tuan and S.M. Chang, "EOG based Human-Computer Interface System Development", Int. Journal Expert System with Applications, Vol. 37, Issue 4, pp: 3337-3343 (2010).
[19] A.B. Usakli, S. Gurkan, Design of a novel efficient human–computer interfacean electrooculagram based virtual keyboard, IEEE Trans InstrumentationMeasurement, 59, 2099–2108 (2010).
[20] L. Y. Deng, C.L. Hsu, T.Ch. Lin, J.S. Tuan and S.M. Chang, "EOG based Human-Computer Interface System Development", Int. Journal Expert Systemwith Applications, Vol. 37, Issue 4, pp: 3337-3343(2010).
[21] M. Merino, O. Rivera, I. Gomez, Al. Molina and E. Dorrenzoro, "A Method of EOG Signal Processing to Detect the Direction of Eye Movements", 2010 First Int. Conf. of Sensor Device Technologies and Applications, pp: 100-105, Italy, Jul. 18-25(2010).
[22] Altman, "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175 185, N. S. (1992).
[23] A. Johnson & D. W. Wichern, Applied Multivariate Statistical Analysis. Prentice Hall (1988).
[24] Greene, William H., Econometric Analysis (Seventh ed.). Boston: Pearson Education. pp. 803–806. ISBN 978-0-273-753568(2012).
[25] Bishop, Christopher M.Pattern Recognition and Machine Learning,Springer,pp. 206–209 (2006).

[26] Mozina, M.; Demsar, J.; Kattan, M.; Zupan, B. Nomograms for Visualization of Naive Bayesian Classifier (PDF). Proc. PKDD 2004. pp. 337–348(2004).
[27] Quinlan, J. R. "Simplifying decision trees". International Journal of Man-Machine Studies. 27 (3): 221(1987)