

# **An Energy Harvesting Portable and Rollable Large Area Gestural Interface Using Ambient Light**

George Quine  
954390

Submitted to Swansea University in fulfilment  
of the requirements for the Degree of Master of Science



**Swansea University**  
**Prifysgol Abertawe**


Department of Computer Science  
Swansea University

December 2021



# Declaration


This work has not been previously accepted in substance for any degree and is not being con- currently submitted in candidature for any degree.

Signed .....  ..... (candidate)

Date ..... 15/12/2021 .....

## Statement 1


This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .....  ..... (candidate)

Date ..... 15/12/2021 .....

## Statement 2

I hereby give my consent for my thesis, if accepted, to be made available for photocopying and inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .....  ..... (candidate)

Date ..... 15/12/2021 .....



*I would like to dedicate this work to science for a sustainable future.  
Also, to my friends and family*





# Abstract

In this work, we explored the viability of using ambient light and a large photovoltaic sheet for energy harvesting and gesture recognition. Our prototype consists of a large, portable, and rollable gestural interface which can uniquely distinguish and classify distinct hand gestures performed by a user. The system works under the principle that the amount of power harvested by the photodiodes decreases when a near-field object blocks the surrounding light. By monitoring these fluctuations, we recognised that different shadow patterns produce a distinct signature in the amplitude of the harvested voltage. Delegating the detection responsibilities to machine learning, it was possible to capture the hidden meaning within the hand gestures to perform an action in real time. We focused on two classifiers, one utilising a machine learning technique, Random Forest (RF), and the other a deep learning classifier, a Convolutional Neural Network (CNN). To further improve the robustness of the system, we applied two pre-processing techniques known as Normalisation and Principal Component Analysis to reduce inherent noise caused by inevitable environmental and human factors. We evaluated the proposed system under a variety of lighting conditions, as well as assessing the significance of the two pre-processing techniques. We trained our models with 1,050 incidents of 5 unique gestures. The CNN demonstrated the highest overall accuracy in all lighting conditions, with 95% accuracy in 1K lux. The RF performed similarly well, obtaining 93% accuracy in 1K lux. Using a designed Graphical User Interface (GUI), both models are capable of recognising an unseen gesture in 0.05 seconds



# Acknowledgements

I would like to express my gratitude to the following people for their assistance with the development and completion of this dissertation.

Dr Deepak Sahoo, my supervisor, for his advice, encouragement, and enthusiasm towards my dissertation.

Swansea University Computer Foundry who have helped me progress from a first-year novice who had never programmed to where I am now.

My parents, Tej and Stuart Quine, for their unfailing support throughout my degree and proofreading my dissertation despite their limited knowledge of computer science.

Last but not least, I'd like to express my gratitude towards my family and friends for believing in me and supporting me during my studies.

# Table of Contents

Chapter 1 Introduction .....	13
1.2 Proposed Method .....	15
1.3 Project Aims and Contributions.....	16
Chapter 2 Related work .....	18
Chapter 3 Project Specification.....	22
3.1 Feature Specification.....	22
3.1.2 Must Have Features of the Prototype by the end of this dissertation .....	22
3.1.3 Optional Features of Both the Prototype and End Product.....	23
3.2 Technology Choices.....	23
3.2.1 InfinityPV Solar Tape .....	23
3.2.2 PicoScope 5000 Series .....	24
3.3 Programming Languages .....	25
3.3.1 Python - Google Colaboratory .....	25
3.3.2 Java – TKinter.....	26
Chapter 4 Proposed Method.....	28
4.1 Gesture Set.....	28
4.2 Recognition Framework.....	29
4.3 Data Collection .....	29
4.4 Pre-Processing of Raw Data.....	30
4.4.1 Principal Component Analysis (PCA) .....	31
4.4.2 Magnitude Normalisation .....	32
4.5 Classifiers .....	33
4.5.1 Random Forest.....	33
4.5.2 Convolutional Neural Network.....	34
Chapter 5 Experimentation of Hyperparameters Using Weights & Biases .....	36
5.1 Convolutional Neural Network.....	38
5.2 Random Forest.....	40
Chapter 6 System Testing .....	44
6.1 Experiment 1: The Impact Light Intensity has on our Final Models .....	46
6.2 Experiment 2: The Flicker Effect .....	49

6.3 Experiment 3: The Importance of Principal Component Analysis.....	50
6.4 Experiment 4: The Importance of Normalisation.....	51
Chapter 7 Discussion .....	54
Chapter 8 Future Work .....	58
Chapter 9 Conclusion.....	60
Chapter 10 Bibliography.....	62



# Chapter 1

## Introduction

Technology has changed the way we operate and behave; we have become more efficient and effective with our time.

*“It has become appallingly obvious that our technology has exceeded our humanity.”*

- Albert Einstein, Scientist

But whilst technology continues to grow, it has been the small gadgets that have made the greatest difference [1]. We have seen new, ground-breaking developments in the wearable technology world. We are witnessing an emergence of an entirely new category of interface systems that carry a profound change in the way we perceive and communicate with technology. Today, smart watches, portable tablets and smart home appliances have changed the way we collect, use, and share data [2][3][4]. As society and technology continue to change, there is a need for a new approach to communicate with these devices.

*“Body language is a very powerful tool. We had body language before we had speech, and apparently, 80% of what you understand in a conversation is read through the body, not the words.”* - Deborah Bull, British Writer

Hand gesture recognition presents a new way for computers to begin to understand human body language. This provides a more efficient pathway for human machine interfacing (HMI) without the need of physical touch. It is believed that gesture-based interfaces can reduce the complexity of communication [5]. Imagine the safety benefits of interacting with your car’s functions without taking your eyes off the road [6]. The ultimate aim is to bring human-machine interfacing (HMI) to a regime where the interaction between man and machine will be as normal as interactions between humans. Therefore, promoting this movement into HMI is an important field of study.

The term gesture recognition refers to the whole process of tracking the performed human gesture, converting the gesture to numerical data, and interpreting the data to represent the meaning of the given command, a non-verbal communication. Many studies have been conducted to track body movements, the most common method being the use of cameras and computer vision algorithms to interpret gestures. The Microsoft Kinect [7] and the Leap Motion Controller [8] are two such examples, both use infrared cameras to gather a detailed analytical view of the user. Although this technique has proven to be very effective, to recognise these gestures applications must invade the user’s privacy and possess the image input from a camera to gain information of the user’s intentions. Other methods studied include contact-based devices which identify gestures by analysing the physical interaction of the user from sensors like accelerometers and gyroscopes. The Nintendo Wii [9] remote and CyberGlove III [10][11] can obtain relatively accurate information due to the direct

attachment with the users, however, a contact-based device can be inconvenient to wear and may require the user to stand in front of a receiving sensor.

As the demand for remote and disposable devices increase, there is growing interest in battery free systems [12]. State of the art gesture recognition systems consume significant amount of power.

Relying heavily on batteries significantly limits their application scope. Replacing batteries or recharging is inconvenient and can be problematic if there is no nearby power resource. Energy harvesting has become an attractive approach to sustain power from different renewable sources, including solar energy, thermal, radio frequency and body movement [13][14][15][16]. Research has already been studied in human gesture recognition involving radio frequency (RF) signals [17]. AllSee, created at the University of Washington, used the surrounding wireless transmissions around us to harvest power and distinguish gestures. Most of their research was based around the concept that the RF signals becomes distorted due to surrounding movement. Their prototype could recognise unique amplitude changes in the wireless signals and accurately identify different gestures more than 90% of the time. However, although RF signals open exciting possibilities for gesture recognition, the energy harvested from RF signals is relatively low. In recent decades, solar energy has been one of the most widely investigated source for energy due to the increasing affordability of the technologies as well as increased political pressure to shift towards green energy sources [18].

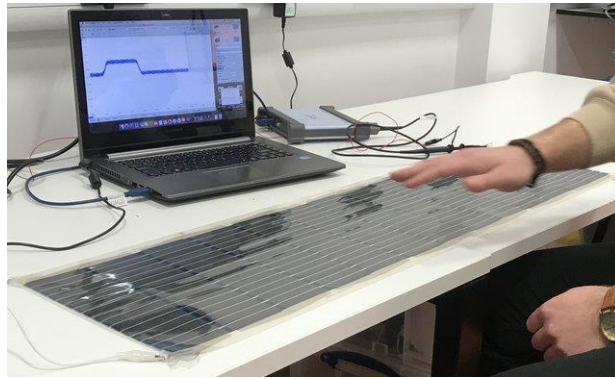
Gesture technology offers multiple advantages over other forms of HMI across multiple applications. In the gaming industry, using gesture recognition in computer games would make the experience more interactive for consumers, especially when combined with existing Virtual Reality systems [19]. Developers have put a lot of energy into the graphics, UX and sound to build a sense of reality.

Although this sense of reality is removed when the user must hold a controller in either hand. This reminds the user that their surroundings are an illusion. Gesture recognition will allow the user to have the freedom of movement without the burden of holding a controller.

Another significant application area is communications technology for those with speech impairments. Sign language is not commonly learnt by people; thus, mute people have problems communicating. Hand gesture recognition could facilitate a gesture-to-speech interpreter to remove the existing communication barrier between deaf and hearing people [20]. The technology also presents applications in industrial settings for use with human robot collaboration. Some factory environments make other HMI system challenging to implement, as noise levels may be too high for voice recognition systems and using touch screen interfaces usually requires a worker to stop his task. Gesture recognition systems would enable workers to control smart robotics and other IoT systems from a distance while still at their station.

## 1.2 Proposed Method

In this paper, we propose a battery-free system that can perform gesture detection and recognition using the surrounding ambient light. Our prototype will consist of a large, portable, and rollable gestural interface which can uniquely distinguish distinct hand gestures from a user. By featuring a rollable photovoltaic sheet, its display size and form factor can be dynamically changed to the user's preference. This enables the system to be highly adaptable with many environments and provides the user a free range of interactions with its portability. The photovoltaic sheet can be rolled out quickly, offering a perfect solution for portable use. The system can be deployed over many large surfaces such as a wall, table, window blind, and cabinet. Additionally, the flexible material enables the system to be deployed over curved or undulating surfaces such as a sofa arm rest. This advancement should be advantageous for consumers where space, size and surface characteristics vary.



*Figure 1 - The Photovoltaic Sheet and Recognition Signal*

Sunlight can help overcome the energy limitations of batteries. The photovoltaic sheet has self-powering capabilities by harvesting energy from the surrounding light. Furthermore, as awareness of the global climate emergency increases, solar energy provides a viable alternative to burning fossil fuels and a pathway to reduce global warming [21]. Photovoltaic (PV) systems have a structure that include the individual photodiodes, storage modules, connections, and additional elements depending on the systems requirements. The most important element of the system, the photodiodes, can directly convert sunlight into electricity by a phenomenon known as the photovoltaic effect. The maximum power generated by the photodiodes increases with the increase in light intensity. This will offer advantages for low-cost installation, portability, and sustainability.

The final system will consist of a large rollable photovoltaic (PV) sheet, an energy harvesting circuit board (InfinityPV OPV3W60V) and a microcontroller board for gesture recognition. As it can be seen in Figure 1, the length and width of the PV sheet is 110 cm and 23 cm, respectively. The photovoltaic sheet can generate up to 12 W at 1 sun or 0.03 W at 300 lux for indoor lighting. We receive a voltage signal which appears as an AC modulation that is filtered by a 1  $\mu$ F series capacitor. This signal will be used for gesture detection.

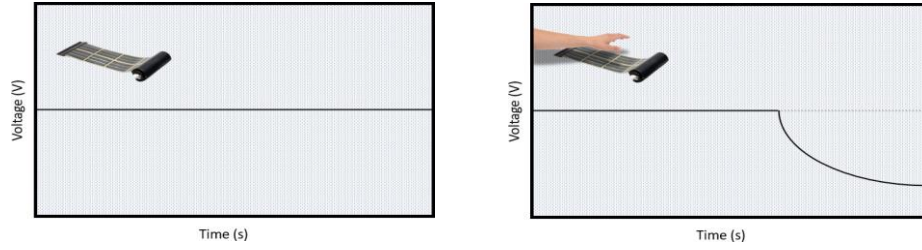


Figure 2 – Time-series of photodiodes harvested power as no hand is placed above the surface interface (left) or hand is placed above surface interface (right).

The system will work under the principle that the amount of power harvested by the photodiodes decreases when a near-field object blocks the surrounding light. As demonstrated in Figure 2, moving a hand above the surface interface in close range will cut off specific light arrays from reaching the photodiodes, causing a sharp dip in the receiving voltage signal. By monitoring these fluctuations, we recognise that different gestures leave distinguishable signatures. Delegating the detection responsibilities to machine learning, we can capture the hidden meaning in the hand gestures to perform an action in real time. Since the advancement of machine learning, we have achieved outstanding achievements in fields such as image and speech recognition and semantic understanding [22]. Without completely depending heavily on human involvement, a machine can achieve superior feature extraction from raw data and discover complex function mapping from very diverse, unstructured, and inter-connected data. This rising territory of machine learning can be introduced within our system to analyse the receiving voltage signal and used to identify different hand gestures. When compared to other gesture recognition technologies, there are several advantages to using light and shadows for recognition. For starters, there are no privacy concerns that camera-based system encounter. From the data received by the system, it is virtually impossible to maliciously create anything from the gathered silhouettes. Compared to contact-based devices, the system can be used completely hands free, resulting in more comfort for the user. Finally, as long as there is light, the system will work anywhere. This is particularly useful for remote regions with no access to any other source of electricity.

### 1.3 Project Aims and Contributions

The desired aims of the projects are as followed:

- To propose the concept of using ambient light to detect and recognise human hand gestures.
- To review light-based HMI applications.
- To briefly review energy harvesting technologies for IoT applications.
- To investigate a series of hand gestures and gather data from a large group of users.
- To review machine learning techniques for gesture recognition.
- To explore different pre-processing techniques. Raw data must be pre-processed before applying the classifier, this will reduce computational time.
- To design a strategy to deal with changing light intensity scenarios.
- To create a system that can work in a large range of different light intensities.
- To test our prototype in lighting conditions and evaluate the accuracy of the system.
- To create a GUI to demonstrate real-time detection for our system.





# Chapter 2

## Related work

In this chapter, we will discuss the existing research papers within the domain of gesture detection, using visible ambient light, and self-powered designs that make use of a corresponding amplitude change in the received energy harvesting signal caused by partly shadowing the solar cells. Throughout the published papers, we will discuss:

- The methodology taken in each paper and how their particular approach works.
- The ultra-low self-powered designs and their approach with energy harvesting.
- The investigated hand gestures used.
- The machine learning techniques explored and how well they performed.
- The testing approach taken in each paper and evaluate the metrics that were used.
- The advantages and the drawbacks that can be improved in the future.

C. Sorescu, et al. investigated a set of six unique hand gestures [23]. The gestures explored were “*a circular swipe clockwise and anticlockwise, opening and closing of the fist, moving the hand up and down, a linear swipe once and a linear swipe continuously*”. A Random Forest (RF) classifier was chosen to identify the above hand gestures based on the resulting analog signals. C. Sorescu, et al. obtained an overall accuracy of 97.2% using 30 collected samples for each gesture using one user. The authors concluded that the clockwise and anticlockwise circular motion shared small variations between the two, leading to a reduce precision and recall value. In other words, both gestures constructed a very similar amplitude change in the received voltage signal. Although for all other gestures, there were no mistakenly predicted false positives or negatives in gesture detection. A RF is a simple yet powerful technique that works on the assumption that a collective opinion of a crowd is smarter than a single individual. The RF is an algorithm that aggregates a large group of individually trained decision trees to work as a committee to make a more accurate prediction. Each individual decision tree is trained in parallel using various subsets of the training dataset, resulting in a large diversity of trees and generally results in improved performance. For the final decision, the RF classifier aggregates the decision of each individual tree and selects the prediction result with the most votes as the final prediction. A RF is invariant to the scaling of the data and requires little effort in the fine-tuning of hyperparameters, yet it maintains a high reproducibility. A RF can almost work out of the box and that is one reason why they are very popular.

Yet, D. Ma, et al. evaluated the performance of multiple machine learning classifiers [24]. This included a Random Forest (RF) and three other classifiers: Support-Vector Machine (SVM), K-nearest neighbours (KNN) and a Decision Tree (DT). Under different light conditions, six user friendly hand gestures were explored, these included: “*Up, Down, UpDown, DownUp, LeftRight and FlipPalm*”. For KNN, with the number of nearest neighbours set to 10 and the distance weighted, KNN performed the most promising results in comparison with the others. At a light intensity of 2600 lux, an environment suitable for both opaque and transparent cells, the authors received 97.1%, 96.5% and 96.1% for S1, T1, and T2, respectively. A KNN algorithm works under the assumption that similar actions will

exist in proximity. When trying to classify a datapoint based on a given dataset, it is compared to its neighbours and classified depending on which points are approximated locally. Once the distance has been computed, the datapoint is classified based on which points share the highest probability. KNN is often called the Lazy Learner, it is very easy to implement as the model does not include a training period. It does not derive any discriminative functions from the training data as the only thing needing to be calculated is the distance between different points. Although, KNN does not work well with large datasets as calculating distances between every data instance would be very costly and degrades performance. M.Kahalokula also decided on using KNN and evaluated how the size of the training dataset and the k-value affects the recognition accuracy [25]. For each gesture subset, the optimal training size was between 8 – 11 when using a k-value between 7-10. Based off these results, their system produced an accuracy of 98% in a dorm room environment.

To minimise the likelihood of falsely extracting gestures that had not transpired, D. Ma, et al. applied a gesture length constraint to detect the start and stop of a gesture [24]. The authors distinguished that 90% of the gestures could be completed within 1 second. Therefore, a length constraint was applied where anything less than 0.2 seconds or greater than 1.4 seconds would be discarded. Previous works used a special preamble scheme to avoid confusion with moving actions in the environment [26]. Although this scheme involved the user to perform two additional hand gestures every time to open the communication channel between the device and the user, this is user- unfriendly. M.Kahalokula used a one-time calibration step to measure the surrounding environment each time before use [25]. During this period, 5 seconds of sensor data is collected, and a median- absolute-deviation (MAD) is calculated. Once any sensor reads an intensity below the calculated median, the system begins to collect a time-series result. Once all the sensors report that their intensities have returned back above the calculated median, the gesture is believed to be over. Although unless the user is in a completely static environment, the surrounding ambient light will change in intensity over time. This calibration step does not respond well with large changes in the surrounding environment and therefore frequent recalibration of the system is necessary.

Y.li et al, presented two prototypes under the domain of gesture detection, applying the concept to a smart watch and smart pair of glasses [27]. Their approach relied on an array of visible small, low-cost photodiodes surrounding the object to harvest energy from the surrounding ambient light and to recognise specific finger gestures in close proximity. D. Ma, et al. the considered the use of transparent solar panels [24]. The authors considered three solar cells with different energy harvesting efficiencies/transparentcies and conducted an experiment to validate the effect the transparency would have on gesture detection accuracy. This experiment involved both a non-transparent silicon-based opaque solar panel (S1 0% transparent) and two transparent organic solar cells (T1 20.2% transparent and T2 25.3% transparent). Transparent solar cells are a new development in solar technology, allowing visible light to pass through, harvesting energy from infrared and ultraviolet light. This new discovery brings a new attraction to solar panels. The transparent solar cell can be fitted on the screen of a smart watch or mobile device to accomplish both energy harvesting and gesture recognition. Their findings validate that transparency does not reduce gesture

recognition as long as the light intensity is above 400 lux. Given that many indoor and cloudy environments are above 400 lux, this suggests that the use of transparent solar cells will not have a negative effect on gesture recognition which offers a more attractive alternative.

Y. li et al conducted a study to measure the amount of energy the two prototypes can harvest in four indoor lighting conditions (200 lux to 2K lux) and three outdoor lighting conditions (4K lux to 110K lux) [27]. In the indoor conditions, the devices were capable of harvesting power in ranges from 23  $\mu$ W to 124  $\mu$ W. Whereas, in the outdoor conditions, the amount of power harvested by the two prototypes were significantly higher, ranging from 1.3 mW to 46.5 mW. The reason for this is because natural sunlight contains more infrared light which can be converted more efficiently into energy. The systems consume 34.6  $\mu$ W/74.3  $\mu$ W for the glasses/watch respectively. Their studies show that the harvested energy is sufficient enough to power the gesture recognition module unless the user is sat in a dark room. This issue can be alleviated by the use of super-capacitors, where 1 – 3 seconds under direct sunlight can provide the system sufficient enough of power to last for one hour in a dark room.

To investigate robustness, both D. Ma et al and Y. Li et al investigated the effect of surrounding interference, specifically the effect of human movement within the background and the impact of varying light intensities [24][27]. This should be carefully looked into depending on where your system would be located. If your system is designed to be used within a busy store to control a screen of display, it is important that the system is still effective whilst the store is busy with moving customers in the background. Y. li et al examined their prototypes in a more rigorous scenario in which the surrounding ambient light fluctuates [27]. The authors investigated the effects of five possible cases including: (1) flashing lights, (2) partial light blockage, (3) nearby body movement, (4) sudden light change and (5) user movement [27]. Both the smart watch and glasses effectively eliminated the high frequency flickering signals, achieving 100% precision and 97-98% recall at 60 and 90 Hz. However, a light flickering at 30 Hz significantly impacted the systems performance. This scenario would be uncommon in any environment since it is noticeable to the naked eye. D. Ma, et al. also investigated the effect of sudden light intensity, their research additionally signified that when light intensity changes very suddenly, accuracy is not affected, whilst when light intensity changes at a low rate almost half of the gestures were wrongly recognised [24]. Next, (2) Y. li et al shaded half of the photodiodes, resulting with one half being under 900 lux whilst the other half was under 400 lux. The precision and recall remained high, 100% and 98% respectively. Someone passing nearby the user in the background, casting a shadow across the photodiodes caused negligible impact on the systems accuracy. This is solely attributed to the photodiodes sensing range being approximately between 0.5cm and 3cm. This was also examined by D. Ma, et al. where Interference only impacted the accuracy by 1.5% [24]. Distanced objects casting shadows from further away block less light and will have less interference. Introducing a sudden increase in light intensity from 550 lux and 800 lux at a rate of 3 Hz had no effect on the performance of the model. Y. li et al system still achieved high precision (96.7% for glasses and 95% for watch) and recall (97% for glasses and 96.3%). Finally, performing the gestures whilst moving through a nonuniform distribution of light ranging from 500 lux to 1K lux, the

system still achieved 100% precision and 97% recall. Only four out of eighty gestures were classified incorrectly.

# Chapter 3

## Project Specification

This section considers the required features of the project in order to meet the projects' goal. The project specification is an important design consideration to ensure completeness and correctness of implementation. The feature set is divided into two categories, the must-have mandatory features and the optional features. The must-have features include a minimum list of requirements the system must have in order for the system to work effectively, whereby the optional features are the features that will enhance the application. This section continues to discuss the language and frameworks that were used to create these features.

### 3.1 Feature Specification

The main objective of the project is to develop an energy harvesting system capable of detecting and recognising distinct hand gestures made by a user using ambient light. Thereby bringing human-machine interfacing (HMI) to a regime where the interaction between man and machine will be as normal and natural as interaction between humans.

Before creating the feature specification, it was important to have a good understanding of what the project required. It was beneficial to concentrate on each various aspect:

- The photovoltaic sheet,
- The energy harvesting capabilities,
- Gesture recognition,
- The connection between the microcontroller and the connected GUI.

Focusing on each aspect has aided in the creation of a list of feature requirements that will accomplish this objective.

#### 3.1.2 Must Have Features of the Prototype by the end of this dissertation

- Light intensity must be directly proportional to the voltage harvested by the photodiodes. The system must dip in voltage once the photodiodes are partially blocked.
- The system must perform gesture recognition using the received voltage signal and a machine learning technique to detect the meaning behind the gesture performed.
- The system must be able to adapt to various conditions, such as varying light intensities.
- The system must be able to detect a gesture in real time (less than a second).
- The prototype GUI must be able to receive the unseen gesture and informally describe the action it had received.

### **3.1.3 Optional Features of Both the Prototype and End Product**

- For the system to require no external power, the photovoltaic sheet must harvest more power than the complete circuit consumes during operation.
- The photovoltaic sheet can be pulled out, retracted and stocked from a plastic case housing. All components will be neatly packed within the housing to improve portability.
- The trained machine learning model will be deployed to a microcontroller allowing for the absence of a connected computer.
- The microcontroller must be able to read the output power harvested by the photodiode system at a high sampling rate.
- The microcontroller must have the ability to connect and share data across a wireless serial communication with a compatible device to perform the gesture action.
- Using a maximum power point tracking circuit, the photovoltaic sheet must effectively charge the lithium polymer battery until fully charged.

## **3.2 Technology Choices**

This section introduces the technology choices used throughout training the machine learning algorithm and the final model deployment phase. We will discuss the languages and frameworks used to implement the requirements of the project.

### **3.2.1 InfinityPV Solar Tape**

InfinityPV Solar Tape features an award winning flexible organic solar cell foil that is capable of powering small-scale niche applications. They do not include toxic or scarce elements which offer a sustainable alternative to traditional solar energy harvesting technologies. The infinityPV solar tape is made up of a thin film of organic semiconductors such as polymers and carbon containing molecules. Since they are largely made from plastic opposed to the conventional silicon, the interest in this material stems with its mechanical flexibility, the principle of being more environmentally sustainable, and the facts that its manufacturing process is less expensive. Additionally, whilst traditional silicon solar panels can weigh between 20 to 30 kilogrammes per square meter, infinityPV solar tape weighs considerably less, between 220 to 450 grams per square meter. Additional figures are shown in Figure 3. [28] [29]

<b>Type</b>	Bidirectional
<b>Width (mm)</b>	110
<b>Thickness OPV (micron)</b>	100
<b>Thickness adhesive (micron)</b>	25 – 50
<b>Thickness tape (micron)</b>	125 – 150
<b>Voltage pr. meter (V)</b>	48 – 52
<b>Current (mA)</b>	40 - 50
<b>Power (mW pr.m)</b>	ca. 1200
<b>Cost (€ pr. m)d</b>	60 (50)

*Figure 3 - Specification of the infinityPV Solar Tape*

### 3.2.2 PicoScope 5000 Series

To train our machine learning algorithm, we shall be collecting our data using an oscilloscope. The PicoScope 5000 series is a USB-powered oscillator which produces oscillating waveform signals with variable frequencies. The frequency of the waveform is varied by the dynamic change in the magnitude of the input voltage. If the input voltage rises, so does the amplitude of the waveform; conversely, as the voltage falls, so does the amplitude of the waveform. Many USB-powered oscilloscopes have real-time sampling rates ranging from 100 to 200 MS/s, but the PicoScope 5000 series has a sampling rate of up to 1 GS/s and a maximum bandwidth of 200 MHz. The digital oscilloscope is supported by an industry-leading software tool, PicoScope 6, which provides powerful effective tools for analysing and collecting data. PicoScope 6 dedicates the majority of its visual display to the waveform to give a clear display of the data recorded allowing us to clearly understand how different gestures made above affect the photovoltaic voltage. In fact, the PicoScope 5000 can capture waveforms 500 milliseconds long with 1 nano second resolution giving us the option to feed our models with more features if necessary. PicoScope6 will enable us to capture the voltage levels produced by the photodiodes over a time series to train our machine learning models. [30]

It is here we see the effect on the harvested voltage produced by the photodiodes whilst different gestures are being performed above. When the user moves their hand above the surface interface in close proximity, specific light rays are being cut off from reaching the photodiodes. Using the PicoScope 5000 series, we can see a sharp dip in the receiving voltage. It is apparent from observing these fluctuations that various movements leave distinguishable signatures in the signal waveform.



## 3.3 Programming Languages

### 3.3.1 Python - Google Colaboratory

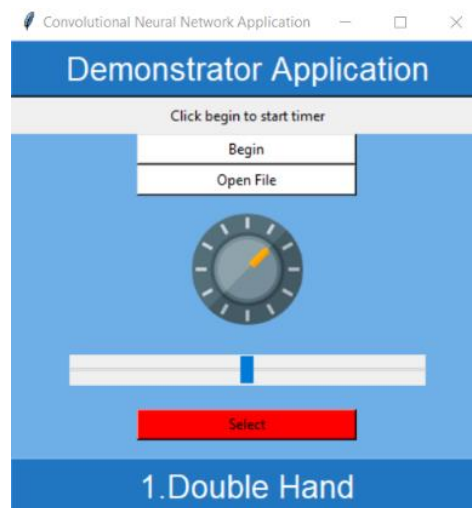
Thanks to its maturing ecosystem of scientific libraries, the Python programming language has established itself as one of the most popular languages for scientific computing, especially in the implementation of machine learning. It is increasingly used not only in academic environments but also in industry. Libraries and services such as Numpy, Scikit-learn, and Google Colaboratory take advantage of this rich environment and provides cutting-edge implementation for a large range of supervised and unsupervised learning algorithms [31][32][33].

- NumPy provides the input data a high-performance framework that helps developers to control their data with powerful mathematical operations before feeding it to the machine learning algorithm. NumPy enables seamless integrations with other scientific Python libraries.
- Scikit-Learn is simple to use, yet it efficiently implements many complex machine learning algorithms for both pre-processing, visualisation of data and evaluation of the final model. Making it an excellent starting point for implementing our models.
- Google Colab is a more advanced python-friendly open-source environment for writing and executing python code through the browser. It is particularly well suited for machine learning because of the minimal needs for setting up and installing packages, while also providing processing capabilities such as GPUs and TPUs for rapid and effective training and running of machine learning algorithms.

Using the libraries and services above will provide a powerful foundation to perform numerical computation whilst training and running our machine learning classifiers for gesture recognition.

### 3.3.2 Java – TKinter

The application graphical user interface (GUI) will be created using TKinter, a built in Python GUI framework. TKinter offers a visual development environment that quickly allows you to design a collection of cross-platform graphical control elements such as buttons, menus and data fields, etc. in a Python application. Once created, these graphical elements can be associated with, or interacted, with features, functionality, methods, data or even other widgets on a computer desktop. TKinter is written in python and is capable of opening and reading files and running trained machine learning models. The GUI will be responsible for declaring the reaction to the UI elements using the trained machine learning model once a classification has been made. [34]



*Figure 4 – A GUI that shows the user what action has been performed in real time*

TKinter will be used to display the different gestures the user performs in real time. Once the action has been received, the gesture will be depicted on the GUI by the movement of the UI elements and some informative text to clearly describe the gestures behaviour. As seen in Figure 4, a circular clockwise/anticlockwise gesture will be depicted by the dial being rotating clockwise or anticlockwise depending on the gesture performed, followed by the text "Clockwise" or "Anti-Clockwise". A left/right swipe gesture is represented by the slider's arm sliding left or right depending on the gesture performed, followed by the text "Left swipe" or "Right Swipe". Finally, a double hand gesture will be represented by alternating the colour of the 'Select' Button from Red to green, followed by the text "Double Hand". This will distinctly demonstrate whether the system works. Once the actions received complements the gestures performed by the user, the system can be enhanced to control more advanced GUIs, such as scrolling through Netflix's movie catalogue on a Smart TV.



# Chapter 4

## Proposed Method

In this section we will be exploiting the effect of different hand movements above the photodiode surface. Here, we introduce the chosen hand gesture set, followed by our recognition frameworks, and the proposed pre-processing techniques used to remove noise from the signal waveform.

### 4.1 Gesture Set

As shown in Figure 5, we explored 5 different hand gestures, these included *a circular swipe clockwise*, *a circular swipe anticlockwise*, *opening and closing of two fists*, *a swipe to the left*, *a swipe to the right*. These gestures were chosen for the prototype with useful functional controls in mind. A circular swipe clockwise could turn the volume up, whereas the circular swipe anticlockwise could turn the volume down. Open and closing of the fist could select the item of choice. Swiping to the left and right could be used to navigate through a horizontal catalogue.



Figure 5 – (1) Circular clockwise, (2) Circular anticlockwise, (3) Opening and closing of two fists, (4) Swipe to the left, (5) Swipe to the right.

Typically, the photocurrent module would produce a stable voltage reading with little noise. Although as explained in section 1.2, moving a hand above the surface interface in close range will cut off specific light rays from reaching the photodiodes, causing a substantial dip in the receiving voltage signal and therefore changing the shape signal waveform of the xy axis, where x represents the time (s) and y represents the voltage (mV). By observing different hand gestures, it was clear to us that these 5 gestures selected above left distinguishable unique signatures within the receiving voltage, as seen in Figure 6. When the signals corresponding to the circular clockwise and circular anticlockwise gestures are compared, they are inverted in time. This may also be seen when comparing the left and right swipe gestures. A double hand gesture, on the other hand, causes the voltage signal to drop and rise sharply with a sustained period of low voltage in between.

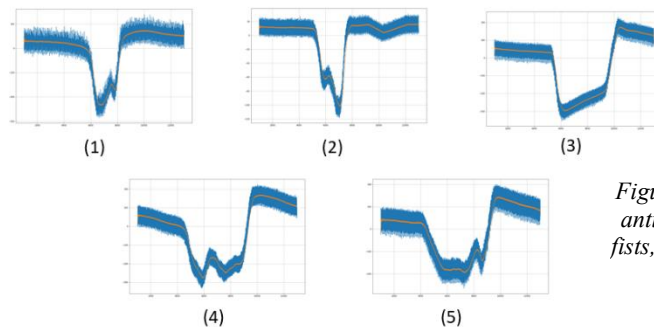


Figure 6 – (1) Circular clockwise, (2) Circular anticlockwise, (3) Opening and closing of two fists, (4) Swipe to the left, (5) Swipe to the right.

## 4.2 Recognition Framework

When the user performs a hand gesture above the photodiode surface, the system captures a time-series of photocurrents and transfers the data to a gesture recognition system in a numerical form. The main efficiency metric for our system is the gesture recognition accuracy. In this paper we are proposing a machine learning based gesture detection system to achieve superior feature extraction from the time series of photocurrents received from the photovoltaic sheet as a voltage waveform. Before we can apply our machine learning models to the raw data, it is extremely important that we pre-process our data.

## 4.3 Data Collection

During data collection, we connected the photodiode interface to an oscillator (PicoScope 5000 series) as shown in Figure 1. Here we carried out the series of gestures listed above in section 4.1. We recorded 15006 photocurrent voltage values over a 5-second time period using the PicoScope and its accompanying data logging software. The data was obtained at a maximum sampling rate of 20k samples per division, with a 500-millisecond collection period and a photocurrent input range of  $\pm 500$  mV. For a single gesture, the photovoltaic module's resultant readings are saved as a CSV file, each containing data separated under two columns, Time (s) and Channel A (mV). The total number of rows in each file is 15009, the top 3 being header rows. As demonstrated in Figure 7, the gesture incidents are then saved into folders, each folder named after the gesture it contains. After that, the gesture-titled folders are stored under the session that they were recorded. Sorting the data will come in handy later during the automation of gesture target labelling, therefore data organisation is essential.

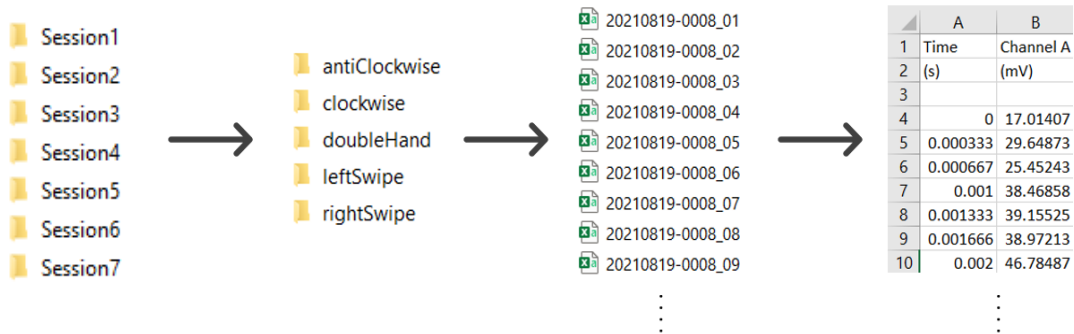


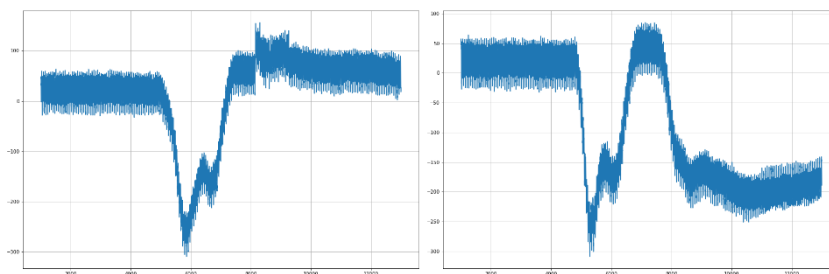
Figure 7- Screenshot of the Organisation of Folders and Files

The Data was collected all by one user using their right-hand. The user performed the task sitting at a desk 2 meters below a ceiling flush light. Data collection was carried in a bright laboratory which measured 750 lux using a digital light meter. Prior to the start of the study, the user was given several minutes to practice the hand gesture before the analysis began. Once the user was confident, the user carried out the 5 hand gestures, repeating each gesture 30 times. In total, we collected 1,050 gesture instances for training and testing of the machine learning models (7 Sessions x 5 gestures x 30 repetitions).

## 4.4 Pre-Processing of Raw Data

We mentioned that our system will be recognising a gesture from the distinguishable signature that the hand gesture leaves in the receiving voltage signal produced by the photodiodes. However, every time a user performs a gesture, the time series data produced by the photodiodes will differ to some degree. Differences in the resulting data may be caused by a variety of cases. 1) Variations in user parameters such as hand size, hand angle, speed of the hand motion, and the proximity of the user's hand to the solar interface; 2) Differences in environmental parameters such as light intensities, for example, turning on/off a light whilst performing an action, or temporary blockage of sunlight due to a moving cloud. Therefore, for the machine algorithm to achieve better results, we must identify and discard the inherent noise in order to minimize the variations before being applied to the classifier. We must pre-process the data to extract the most important features to achieve a better representation of a distinct gesture signature. If the data is pre-processed correctly, it will lead to an increase performance accuracy for the gesture recognition system.

First, the top three header rows were eliminated from each dataset. These columns were redundant and should not be passed through to the machine learning models. This left each gesture dataset with 15,006 numeric values, with the first column being Time (s) and the second column being Channel A (mV). Second, I plotted the photocurrents onto an xy graph to measure the gestures trend of direction across the 5 seconds of recording. X represents the Time (s) and y represents the voltage recorded (mV). For a lot of machine learning applications, it helps to be able to visualize your data. Doing so I could visually identify any irregularities within the data points and remove any gesture that completely deviated from the common trend. Only 2 gestures were completely removed from the dataset and rerecorded. With one example given to the right of Figure 8, both gestures had no resemblance to the common pattern and were the consequence of a recording error. To accommodate for real-world events, tiny blips such as the gesture to the left of Figure 8, were ignored and remained within the dataset.



*Figure 8 - Irregularities within the data. Left was kept in the dataset to accommodate for real-world events. Right was removed from the dataset completely*

The number of datapoints must be reduced in order to reduce both memory and execution time. As it can be witnessed in Figure 9, It was evident that for every individual incident of a gesture the graph presented a stable voltage reading before and after the gesture had been performed. This was a result of the user's hand not being present above the solar panel and the gesture had yet to be performed. I realised I could remove extraneous datapoints by reducing the time range from both the start and end of a gesture. Because all gestures must contain the same number of features, it was critical that the new time range was broad enough to encompass the start and end of all gesture signatures while being small enough to remove superfluous datapoints. In addition, I extracted the estimated weighted mean with a span of 200 to significantly reduce noise whilst preserving the gestures direction. As a result, after optimising the time range and extracting the estimated weighted mean, we were able to delete 3,006 superfluous datapoints from each unique gesture. This left each gesture with 12,000 voltage datapoints from its original 15,006, resulting in a significant reduction in memory and computational time (3,006 superfluous datapoints multiplied by the total of 1,050 gesture incidents used for training is 3,156,300 datapoints that have now been removed). Next, all gestures were concentrated to one single data frame. For each incident of a gesture, the 12,000 datapoints are presented as a new row horizontally with the last column being the true targeted gesture that we are attempting to predict. This is why it was critical to organise each gesture incidence under suitable file names when collecting data. A python script was used to automatically extract the gesture's datapoints and the resulting class name by reading the folder name in which the gesture was located.

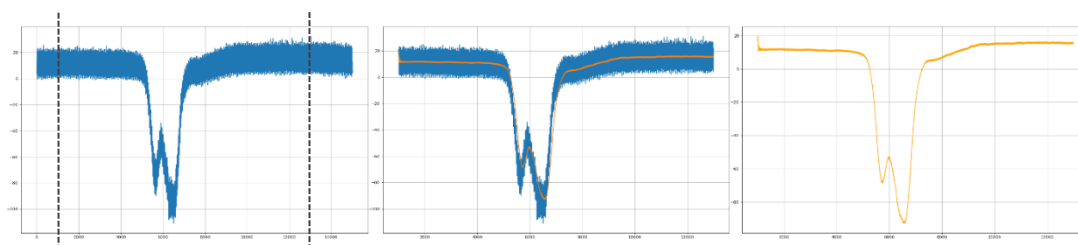


Figure 9 – The removal of extraneous data. First, the data was cropped. Second, the Estimated Weighted Mean was Calculated. Third, noisy data was removed

#### 4.4.1 Principal Component Analysis (PCA)

The label column was removed from the training dataset and prepared for Principal Component Analysis (PCA), an unsupervised dimensionality reduction technique. High dimensional data can pose problems for machine learning as predictive model may run into the risk of overfitting. Dimensions are nothing but features that represent the data. Our data has 12,000 voltage datapoints per gesture, these are the dimensions or features that represent a single gesture. Feeding our machine learning models with this many datapoints will lead to a very slow performance and poor accuracy. PCA is the most used and most popular statistical method for transforming attributes of a dataset into a new set of uncorrelated attributes known as principal components (PCs) [35]. PCA's main goal is dimensionality reduction and feature selection, taking datasets with many features and simplifying that dataset by capturing the principal components that hold the most variance to summarize the data using less properties. PCA is performed by eliminating insignificant features from a high-dimensional space, and projecting the most important features into a lower-dimensional subspace, improving classification accuracy and reducing

computational time. PCA projects the data onto a new space by compressing the most important information onto a subspace with lower dimensionality than the original space. It is a powerful data representation method, being able to capture the most variable data components of samples and extracting important features [36]. In this paper, we wanted to apply the minimum number of principal components such that 99% of the variance was retained. A sample output of PCA is shown in Figure 10, with cumulative explained variance sketched as a function of the number of principal components. The ‘significant dimensionality’ of our desired subspace is defined as the number of uncorrelated principal components that can explain 99% of the variance of features. The way this graph is utilized is by drawing a line across from 99% variance and down to show how many features are required. We discovered that 15 principal components are the bare minimum for retaining 99.007% of the variance in the dataset. In other words, using PCA we have reduced 12,000 datapoints for each gesture to 15 PCs without compromising the information extracted from the data. Finally, the new Principal Components are assigned to its resulting label.

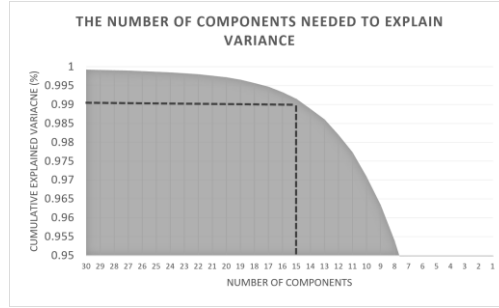


Figure 10 - The number of components needed to explain variance

#### 4.4.2 Magnitude Normalisation

Next, the datasets must be calibrated such that all values correspond to the same scale in order to compare different gestures over varying time-series. The magnitude of the voltage produced by the photodiodes will most likely change between gestures as a result of a combination of human and environmental factors such as the distance between the user's hand and the photovoltaic interface or gestures performed under varying light intensities. Therefore, for every gesture, we will apply a MinMax scaling technique to the voltage magnitude and map the features between the range of 0 and 1, where 1 denotes the maximum magnitude value and 0 denotes the minimum magnitude value.

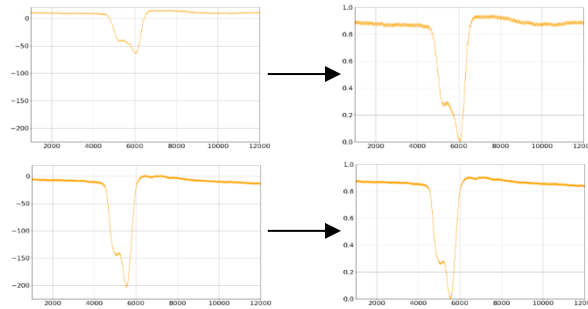


Figure 11- (Top) The application of Normalisation to a gesture performed at 350 lux. (Bottom) The application of Normalisation applied to a gesture performed at 750 lux



In Figure 11, the above two graphs were captured at a light intensity of 350 lux, whereas the bottom two graphs were captured at a light intensity of 750 lux. The results prior to normalisation are on the left, and the results after normalisation are to the right. The magnitude of the two light intensities differs dramatically, with 350 lux only capable of producing a photodiode input range of 20 to -65. Whereas, at 750 lux, the signal range is much larger, ranging from 20 to -205. Applying normalisation will guarantee that all features are in the exact same voltage magnitude scale, allowing the classifier to easily compare different gestures performed under different intensities.

Next, the dataset was shuffled to eliminate any elements of bias patterns within the dataset, thereby improving training quality. The label column was removed and set aside from the training dataset. A 20% subset of the dataset was cut to serve as a validation dataset. A validation dataset is a subset of the training data and is used to provide an unbiased evaluation of a model whilst training, allowing us to tune our model hyperparameters effectively. From the total 1,050 gesture incidents recorded for training. After the split, the training dataset will contain 840 gesture incidents and the validation subset contains 210 gestures.

## 4.5 Classifiers

In general, the effectiveness and efficiency of a machine learning model is often dependent by the nature and characteristics of the data as well as the performance of the learning algorithm. Thus, engineering your data and selecting an appropriate learning algorithm that is suitable for the application is critical and can be challenging. Classification is the action of feeding some features into a machine learning model and requesting it to assign those features to a class or category. During supervised learning, a collection of labelled training data (input data and its resulting output target) is fed into the machine learning model in order to build a function that maps an input to an output based on example input-output pairs learnt from training. Once the model is trained, the model will be able to accurately assign an unlabelled and unseen gesture incident to a specific category. For this project we will be comparing one commonly used classifier against a deep learning machine learning method for gesture recognition in order to determine which model is best suited for my application. From my background research conducted in section 2, both the Random Forest (RF) classifier and the Convolutional Neural Network (CNN) look suitable for my application. Using the 1,050 gesture instances obtained throughout the 7 sessions, we will evaluate the two models by comparing their performance results.

### 4.5.1 Random Forest

A Random Forest is a simple yet powerful technique that works on the assumption that a collective opinion of a crowd is smarter than a single individual. The Random Forest, as the name suggests, is an algorithm that aggregates a large group of individually trained decision trees to create a 'forest' that will act as a committee to make a more accurate prediction. Each individual decision tree is trained in parallel using various subsets of the training dataset, resulting in a large diversity of trees which generally results in improved performance. For the final decision, the RF classifier aggregates the decision of each individual tree and selects the prediction result with the most votes as the final prediction. A

Random Forest is invariant to the scaling of the data and requires little effort in the fine-tuning of hyperparameters, yet it maintains a high reproducibility. A RF can almost work out of the box and that is one reason why they are very popular [37][38].

### 4.5.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are computing models that were first inspired by the biological neural network that constitutes the human brain. A human brain consists of  $10^{15}$  connections and roughly the same number of neurons as there are stars in the Milky Way, around 100 billion [39]. The brain can learn by changing, removing, or forming these connections, similar to how a convolutional neural network can gradually adjust the weights of connections between artificial neurons. Convolutional neural networks are used in a wide range of modern AI technologies, particularly in the machine processing of image processing sets, but also in sequential data.

A CNN is a deep learning machine learning method that is based on the artificial neural network (ANN). The most significant advantage of CNN over traditional approaches, such as the Random Forest, is its unique ability to automatically extract major features for classification throughout the learning process, thereby eliminating the need for such fixed and hand-crafted implemented features. The most commonly used CNN is the 2 dimensional-CNN which is best suited for pattern recognition and image classification. Inherited from the 2D-CNN is a newly emerged 1 dimensional-CNN which has been proven to be very effective in extracting features from 1D sequential time-series data. The primary difference between the two models is the structure of the input data and how the convolutional filter traverses through the data to extract features. Prior to the introduction of 1D-CNNs, 1D signals would have had to be explicitly converted into suitable 2D image formats before being fed into a conventional 2D-CNN. This conversion of 1D signals into image results in a loss of information, which makes 1D CNNs superior in cases of 1D signals. Due to the one-dimensional nature of our data, the proposed model will be a 1D-CNN, where the input layer will receive the raw 1D signal before passing it forward towards the two convolutional layers (), a fully connected layer, and a SoftMax activation function to determine the final output of the node. The output for the model will be a 5-element vector containing the probability of a given window belonging to each of the 5 gestures. [40] [41]



# Chapter 5

## Experimentation of Hyperparameters Using Weights & Biases

Hyperparameters play a vital role in determining a machine learning model's performance. In this section, I will be using the Weights & Biases tool to search through the hyperparameter space to find the most optimal model for both the Random Forest and the Convolutional Neural Network.

The terms hyperparameters and parameter are frequently used interchangeably, but there is a distinction between the two. The parameters of a model are the properties of the training data that the machine learns based on the dataset, such as the weights and biases in the neural network. The model's parameters are not set manually and will vary differ for every experiment. The model hyperparameters, on the other hand, are defined manually by the developer prior to training, for example, the number of neurons within a hidden layer.

The performance of the machine learning model is driven by hyperparameters, so determining the most effective combination of hyperparameters is critical. When a model is correctly implemented with the correct hyperparameters, you can unlock the true maximum potential of your model. It is not possible to determine the best value for a hyperparameter right away and therefore experimentation is required. For example, if the learning rate is too low, the model will miss the important patterns in the data and become stuck. If it is too high, it may cause the model to converge too quickly to a suboptimal solution.

For my project I shall be using Weights & Biases to conduct a hyperparameter search. Weights & Biases is a Python package for tracking, comparing, and visualising hyperparameters in order to discover the most powerful model. Tuning the model is the process of experimenting with different combinations of hyperparameters to find the most optimal architecture. The hyperparameters that I will be tweaking for my project using Weights & Biases are:

### Convolutional Neural Network

- The number of epochs (epochs), in the range of 10 to 400.
- The number of neurons within the second hidden layer (cnn\_1), in the range of 10 to 2,000
- The activation function of the hidden layer (Optimizer), with values Adam, SDG and AdaBound
- Finally, the learning rate (learning\_rate) with a range of 0.001 to 0.1

## Random Forest

- The number of trees to be used within the forest (`n_estimators`), in the range of 5 to 500,
- The maximum depth of a tree (`max_depth`), in the range of 1 to 100,
- The function used to measure the quality of a split (`criterion`), with values Gini and Entropy.
- The minimum number of samples required to be at a leaf node (`min_samples_leaf`), in the range of 1 to 10.
- And the minimum number of samples required to split a node (`min_samples_split`), in the range of 2 to 50.

When commencing an experiment, a small batch size of 8 is a good starting point and should be recommended. A batch size of 8 means that the model will process 8 samples from the training dataset before computing an error gradient and updating the models' inner parameters. A batch size of 8 can be considered as a mini batch, because the value is greater than one and less than the total number of samples in the training dataset. During training and experimentation, I shall be using the validation subset to evaluate both model's optimal hyperparameter architecture. The validation subset contains samples of unseen data held back from the original training data. It is used to provide an unbiased assessment of a model effectiveness on the training dataset while tuning the models hyperparameters.

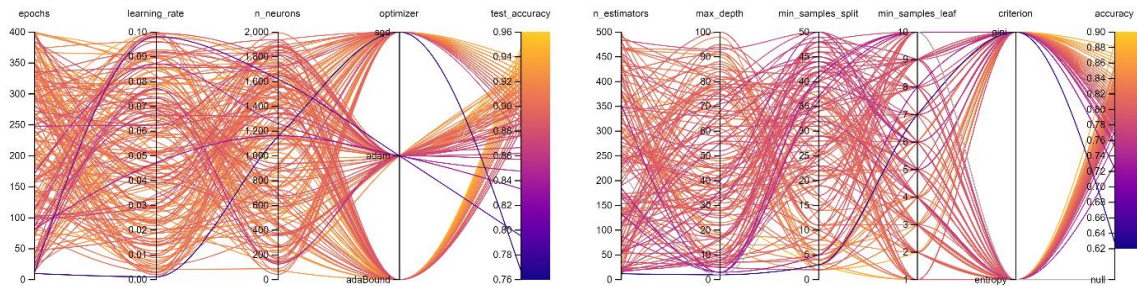


Figure 12 - The Parallel Coordinates of both models hyperparameters. CNN (left) and RF (right)

To begin, I used Google Collaboratory's NVIDIA Tesla T4 GPU to train both models [33], which is an online cloud based Jupyter notebook environment. Thanks to the enhanced computational speed, I ran over 100 different combinations of hyperparameters to determine the most powerful model. In Figure 12, we can visualise the relationship between different hyperparameters and the resulting final accuracy of both the CNN and RF. Both models were evaluated using the 20% validation dataset and performed well, achieving a maximum accuracy of 94.4% (CNN) and 88.1% (RF) prior to tuning.

Using Weights & Biases hyperparameter importance panel, we can visualise the most important features as well as the features that correlate to a high accuracy. The distinction between importance and correlation is that importance accounts for the interaction between the different hyperparameters, whereas correlation accounts for the interactions between the different hyperparameters against the metric accuracy. The importance column shows us the degree to which each hyperparameters was useful in predicting the chosen metric accuracy. The correlation column depicts the linear relationship between the hyperparameters

performance and the accuracy of the model. Correlation values range from -1 to 1, with negative values indicating a negative linear correlation relationship, positive values indicating a positive linear correlation relationship, and a value of 0 indicating no correlation whatsoever.

## 5.1 Convolutional Neural Network

We can see from the hyperparameter importance panel in Figure 13 that the learning rate, epochs and number of neurons are all important features, thus the subsequent optimizers can be fixed first and be given a value so that we can focus on the most relevant hyperparameters. Although the optimizer ‘SGD’ is considered more important, I will be using the optimizer ‘AdaBound’ because of its greater positive correlation relationship with accuracy.

Adabound was introduced in 2019 with the purpose of bridging the empirical gap between Adam-like techniques and SGD by combining the advantages of both optimizers. The AdaBound optimizer employs dynamic bounds on learning rates to achieve the goal of transitioning from the adaptive optimizer to the SGD optimizer. Adabound has demonstrated to address the ability to handle SGD’s slow convergence speed and Adam’s generalization ability. More critically, it requires less memory. [42]

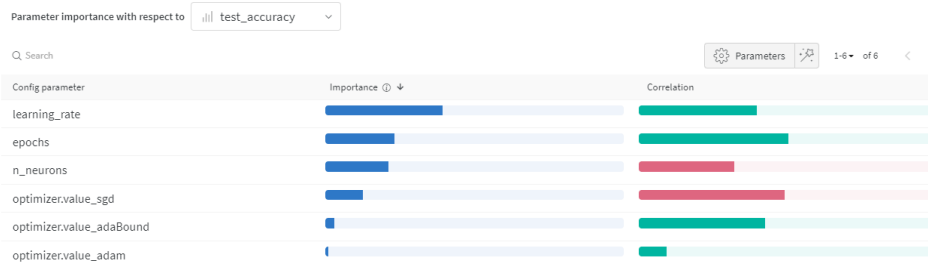


Figure 13 - Hyperparameter Importance Panel of CNN

The main objective of a machine learning model is to minimize the loss function in relation to the models parameters. A low loss function implies that the model can learn well with the dataset provided. It is used to calculate the error between the neural network's output and the true goal value. After fixing AdaBound as the optimizer for the convolutional neural network, the model was run another 100 times to see how the more important features behaved after this modification. The number of epochs is now showing to be of least importance.



Figure 14 - How the number of epochs effects the performance accuracy and loss of the model

An epoch is the number of times a whole dataset is passed through the neural network model. One epoch signifies that the training dataset has been processed through the neural network once. Underfitting can occur when the number of epochs is insufficient, as the neural network will not have had enough of an opportunity to learn. On the other hand, too many epochs will result in overfitting, in which the model can predict the training data very well but fails to predict new unseen data well enough. Looking at Figure 14, we can see that around 120 epochs the model performs at its best before reaching a state of plateau where little growth or decline occurs within the accuracy. Choosing an epoch value greater than 120 will consume unnecessary computational burden to my model, slowing the rate of prediction. With the loss function decreasing, it is also evident that the model can learn extremely quickly, and the training process is going well. If the epoch range were to be extended, I would expect the accuracy to continue converging and the loss function to begin growing in a bowl shape. This is a sign of overfitting.

Continuing the models training with the optimization value set to AdaBound and the number of epochs set to 120. After 100 runs, the number of neurons is next to be the least important hyperparameter. The number of neurons should be adjusted to the model's complexity. Usually, a more complex task requires more neurons to extract the datasets significant features. Our convolutional neural network has two fully connected convolutional layers, the first with the same number of neurons as the number of inputs, and the second were yet to decide. To determine the number of neurons for the second convolutional layer, we looked at how neurons affected both accuracy and loss. What we can see in Figure 15 is that the model performs best around 400 neurons before declining in accuracy with the loss function converging at a reasonable value.

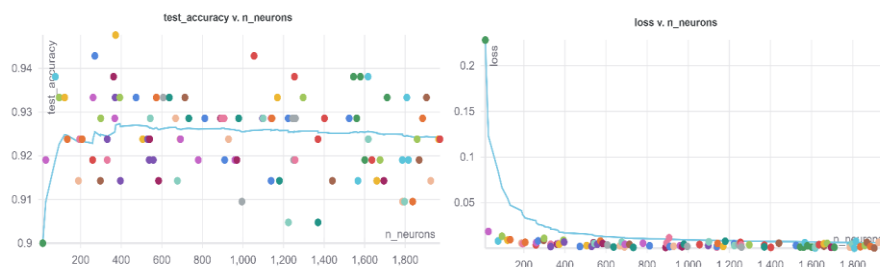


Figure 15 - How the number of neurons effects the performance accuracy and loss of the model

Now that we've set the optimizer to AdaBound, the number of epochs to 120, and the number of neurons to 400, we're down to our final hyperparameter, the learning rate. By fixing the above hyperparameters and running the model 100 times, we can determine the best value for our learning rate.



Figure 16 - How the learning rate effects the performance accuracy and loss of the model

The learning rate is perhaps the most important hyperparameter and as expected left until last to tune. The learning rate is a hyperparameter that defines the adjustment in weights each time the weights are changed during training. Large learning rates cause large weight changes and small learning rates cause small weight changes. If the learning rate is 0.0, the network will not learn and therefore the range is often set between 0.001 and 0.1. It is crucial to explore how the learning rate affects the model's performance. A value too small may result in a long training process, making very tiny updates to the weights in the network that could get stuck, whereas a value too large may result in learning an optimal set of weights too fast resulting to a failure to train. When entering the optimal learning rate zone, you'll observe a quick drop in the loss function before converging. Looking at Figure 16, we see the steepest decline in loss is between 0.015 and 0.020 with an increase in accuracy. Shortly after, we see a deficiency in accuracy. This could be due to the learning rate being too large, causing the weights to never settle down, leading the algorithm to constantly overshoot its aim and 'unlearning' what it has learnt. If we had used a learning rate range of 0.001 to 10, I believe the model's accuracy would have declined more dramatically on the graph. As a result, I am confident that the model will perform well with a fixed learning rate of 0.0175.

After carefully tweaking each hyperparameter, in order of least importance, I have come to a final structure for my convolutional neural network models architecture. On the validation subset, the model attained an accuracy of 96.19%, which is 1.79% greater than before the hyperparameter evaluation. The final structure will involve:

- The Optimizer set to AdaBound,
- 120 epochs,
- 400 neurons for the second convolutional layer,
- And a learning rate of 0.0175

## 5.2 Random Forest

Moving on to my Random Forest architecture, we have 6 key factors affecting the algorithms performance. We originally see from the hyperparameter importance panel in Figure 17, that the minimum number of samples required to split an internal node, the maximum depth of a tree, the minimum number of samples required to be at a leaf node, and the total number of trees are all very important hyperparameters and correlate strongly to a high accuracy. As a result, the succeeding criterion hyperparameter can be fixed first and assigned a value, allowing us to focus on the most important hyperparameters.

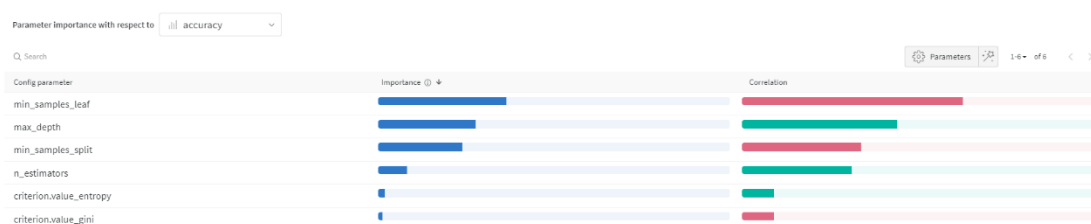


Figure 17 - Hyperparameter Importance Panel of RF



Entropy outperforms Gini in terms of accuracy, as evidenced by the parameter importance panel, hence Entropy will be the model's first fixed hyperparameter. A decision tree's hierarchical structure guides us to the final outcome by traversing through the tree's nodes. A decision tree makes decisions by splitting nodes into sub-nodes. This process is performed multiple times during the training process until only homogenous nodes are left. A process called node splitting is a key concept of Random Forest, it is the process of dividing a node into multiple sub-nodes to create a node more pure than its parent. Both Entropy and Gini Impurity are used as decision tree selection criteria. In the random forest, these two indicators are used to determine where an appropriate split point for a decision node is.

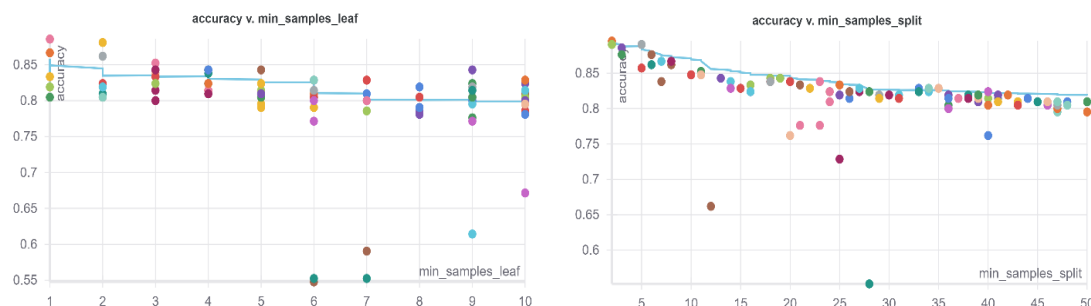


Figure 18 - The effect of `min_sample_leaf` and `min_sample_split` on the performance accuracy of the model.

Continuing the models training with the criterion value set to Entropy. After 100 runs, both the minimum number of samples required to be at a leaf and to split an internal node both constitute to a substantial strain on the accuracy if not set correctly. In Figure 18, we see a decline in accuracy from the set go. The Random Forest default values for `min_sample_leaf` and `min_sample_split` are 1 and 2 respectively. It is crucial to know the difference between a leaf and an internal node in order to comprehend these two hyperparameters. An internal node will contain more splits, whereas a leaf is a node that has no further leaves. The number of samples in the node will be evaluated by the `min_samples_split` parameter, and if it is less than the minimum, no additional splits can be performed at this node, and the node will become a leaf. Whilst the `min_samples_leaf` parameter, on the other hand, checks before the node is formed to see if the possible split results in a child with less samples. If the child results in less samples than specified, the split will be rejected. Increasing the `min_sample_leaf` hyperparameter can avoid the tree growing too deep, but this may result in under-fitting. It was interesting to explore these two hyperparameters, however we'll leave them at their default values.

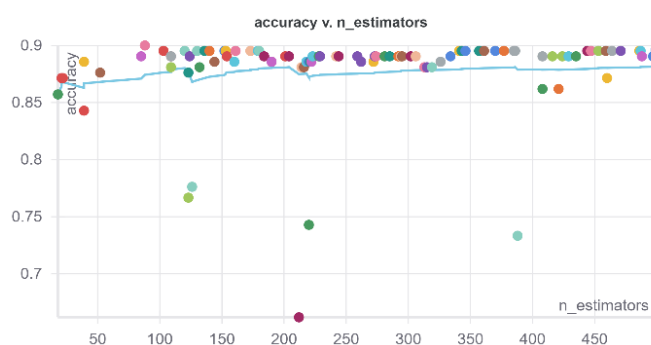


Figure 19 - How the number of trees effects the performance accuracy of the model

Now that the criterion has been set to Entropy, the min samples leaf has been set to 1, and the min sample split has been set to 2. As we continue our hyperparameter experiment, we see that the number of trees in our model is the next parameter to be fixed. Because the Random Forest approach is an ensemble modelling technique, it 'increases generalisation' by generating a range of different types of decision trees on different subsamples of your dataset with different depths and diameters. This hyperparameter indicates the number of trees we want to build in our Random Forest before taking the maximum voting or averages of predictions to conclude our final classification outcome. Looking at Figure 19, we see from our data that the model performs at its best in terms of accuracy when the number of trees is set to 120. Choosing an `n_estimators` value greater than 120 will no longer provide an advantage but consume unnecessary computational burden to my model, slowing the rate of prediction.

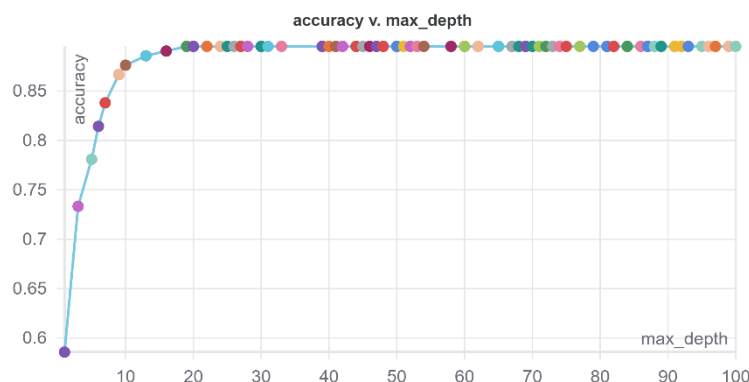


Figure 20 - How the max depth of a tree effects the performance accuracy of the model

With all hyperparameters fixed, we are now left to tune the max depth of our random forest trees. The max depth of a tree is defined by the longest path between the root node and the leaf node. The deeper the tree, the more splits it will have to capture the data's information. Shallow trees, also known as weak learners, are expected to perform poorly since they capture few details about the problem. Deeper trees capture too many aspects of the problem and overfit the training dataset, making it difficult to make accurate predictions on new data. We can see in Figure 20 that the model reaches its optimal performance at a max depth of 20 before reaching a plateau.

After carefully tweaking each hyperparameter, I have arrived at a final structure for my Random Forest. On the validation subset, the model achieved an accuracy of 89.52%, which is 1.42% greater than before the hyperparameter experimentation. The final structure will include:

- The number of trees to be used within the forest to be fixed to 120,
- The maximum depth of the tree to be fixed to 20,
- The function used to measure the quality of a split to be fixed as Entropy.
- The minimum number of samples required to be at a leaf node to be fixed to 1,
- And the minimum number of samples required to split a node to be fixed as 2.



# Chapter 6

## System Testing

In this section, we will assess the performance of our final two systems in relation to a variety of environmental conditions. In order to demonstrate the significance of both pre-processing techniques, we shall also evaluate both models with and without PCA and Normalisation. All evaluations will be performed on Google Collaboratory's NVIDIA Tesla T4 GPU [33].

To enclose all the relevant information about each algorithm we shall be presenting our results in the form of a confusion matrix. A confusion matrix is a cross table that records the number of occurrences between two rates, the true value and predicted value. The columns represent the predictions of the model, while the rows represent the true actual value. The gesture classes are displayed in the rows and columns in the same order; therefore, the correctly classified elements are positioned in the main diagonal from top left to bottom right, and they correspond to the number of times the model correctly identifies a gesture successfully. Apart from helping us compute important quality metrics, confusion matrices are also useful to understand which gestures are being mistakenly classified the most.

In our model, the most important indicator to measure the performance of a gesture detection system is the accuracy, which is defined as the percentage of correctly identified gestures verses the total of gestures performed. In addition to this, we will also evaluate the precision, recall and f-score of both models. These quality metrics can be calculated by using the following notions:

- True Positive (TP): The number of gestures that have been predicted correctly by the model and match the actual value (The actual value and predicted value are the same).
- False Positive (FP): The number of gestures that have been predicted incorrectly by the model in regards to the predicted value, and were falsely assigned to the gestures data set by the model (The sum of the corresponding column except the TP value).
- False Negative (FN): The number of gestures that have been predicted incorrectly by the model in regard to the actual value, and were not recognised as belonging to the gestures data set by the model (The sum of the corresponding row except the TP value).
- True Negative (TN): The number of gestures the model successfully predicted as incorrect and successfully recognised that the gesture does not belong in the gestures data set (The sum of all columns and rows except the values of that class that we are calculating the values for).

We will compute our quality metrics independently for each class and then take the average, hence treating each class equally to show the overall performance of the model. With the notions listed above, I will be using the following measures to evaluate the total number of correct classifications:

- Accuracy: The percentage of correctly identified gestures verses the total of gestures performed

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: The percentage of correctly classified gestures penalised by the number of incorrect classifications (e.g., the fraction of the total correctly identified clockwise gestures by the total number of gestures that the model believed to be clockwise). In other words, this quality metric tells us how much we can trust the model when it predicts a gesture.

$$Precision = \frac{TP}{TP + FP}$$

- Recall: The percentage of correctly classified gestures penalised by the number of missed entries (e.g., The number of correctly predicted clockwise gestures out of the total number of clockwise gestures within the data set).

$$Recall = \frac{TP}{TP + FN}$$

- F-Score: Measures the harmonic mean of precision and recall, which serves as a derived effectiveness measurement. F-Score provides an indication of overall classification accuracy as a weighted average of precision and recall for a specified confidence threshold.

$$F - Score = 2 \times \left( \frac{Precision \times Recall}{Precision + Recall} \right)$$

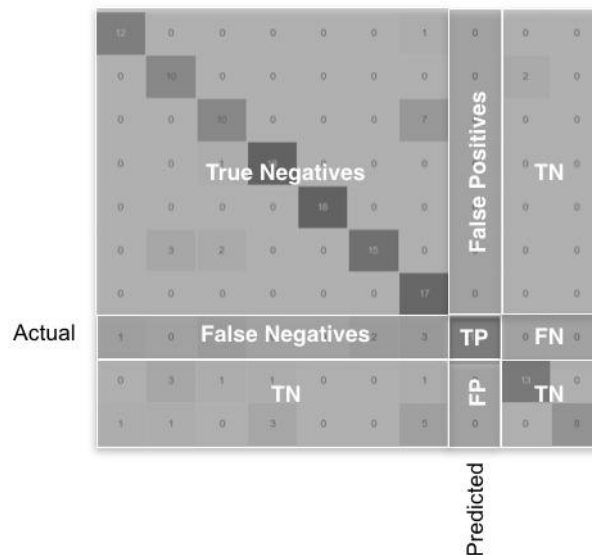


Figure 21 - A Confusion Matrix and its resulting notions

## 6.1 Experiment 1: The Impact Light Intensity has on our Final Models

Both our models rely on ambient light to identify the gesture being performed above. Here we will evaluate both models' robustness across 6 unseen different light intensities, including a near window (1K lux), a bright lab (750 lux), a normal office (650 lux), a kitchen (450 lux), a living room (350 lux), and a dark room (250 lux). In each of the tested light conditions, we have collected 150 gesture instances (5 gestures x 30 repetitions). These light intensities were measured using a digital light meter directly above the surface of the photovoltaic sheet. All samples were taken by one user. Table 1 and 2 shows the results of all tested conditions with their accompanying evaluation metrics.

CNN				
Evaluated Dataset (lux)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
Validation Dataset (750)	96	96	96	95
Near Window (1K)	95	96	95	95
Bright Lab (750)	94	94	94	94
Normal Office (650)	83	83	83	83
Kitchen (450)	71	67	71	67
Living Room (350)	47	48	47	44
Dark Room (250)	31	26	31	28

Table 1- Evaluation Table for CNN

RF				
Evaluated Dataset (lux)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
Validation Dataset (750)	90	90	90	89
Near Window (1K)	93	94	93	93
Bright Lab (750)	86	88	86	86
Normal Office (650)	80	81	80	80
Kitchen (450)	67	62	67	63
Living Room (350)	40	45	40	40
Dark Room (250)	26	23	26	22

Table 2 - Evaluation Table for RF

The CNN evaluation measures are depicted in Table 1. The model demonstrated excellent precision and recall in unseen light levels surpassing 750 lux. This came to no surprise given that the dataset used to train the model was collected at 750 lux. When placed near a window at 1K lux, the model performed at its best, obtaining 96% precision and 95% recall. At high light intensities, the photovoltaic sheet could compute clean clear signal waveforms when a gesture was being performed. This makes it easier for the model to distinguish between gestures, which is useful for gesture recognition. The model performed just as well on the gesture collection performed within the same bright lab as the model was trained with (750 lux). This indicates that the careful tweaking of hyperparameters in Section 5.1, we were able to determine the most powerful combination of hyperparameters, avoiding both overfitting and underfitting and therefore unlocking the true potential of our CNN model.

When the model was performed in a normal office scenario (650 lux), we see a deficiency in precision (79%) and recall (78%). According to our confusion matrix for 650 lux in Figure 22, a right swipe is the first of the gestures to become difficult to classify and is frequently misinterpreted as a double hand gesture. As we continue to decrease the light intensity to 450 lux, the system now anticipates all right swipe actions as a double hand gesture. Left swipe also becomes harder to distinguish and is commonly mistaken as a right swipe gesture, whilst for all other classes, the model continues to perform flawlessly. All 30 clockwise gestures were classified correctly, and only 1 or 2 anticlockwise and double gestures were classified incorrectly.

More than half of all gestures classified below 350 lux are now inaccurate, the model now has an accuracy of 42% which is unacceptable for a gesture detection system. Both anti-clockwise and clockwise signatures are becoming increasingly difficult to distinguish between the two, with 24 of the total 30 anticlockwise gestures being incorrectly recognised as clockwise. Although, the reduced light intensity have yet to hamper the detection accuracy of the double hand action, it continues to perform flawlessly, with only one wrong classification out of a total of 30 double hand gestures. With a light intensity of 250 lux, the model performs incompetently with an accuracy of 31%. In low light conditions the light intensity is not sufficient enough to create substantial signatures in the receiving voltage.

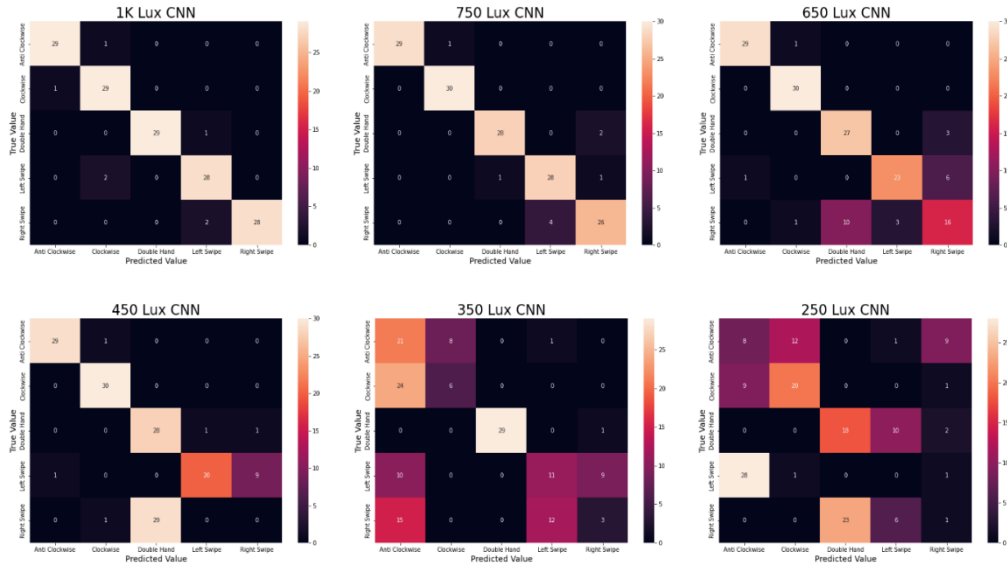


Figure 22 - Confusion Matrix for CNN in various light intensities

Meanwhile, Table 2 depicts the RF evaluation measures. The RF model also exhibits remarkable precision and recall in unseen light intensities over 750 lux, demonstrating that the hyperparameters chosen in Section 5.2 are the model's most optimal combination, avoiding underfitting and overfitting of the training dataset. The model outperformed the validation dataset in all quality metrics when recorded near a window at 1K lux. This is most likely because high intensity light guarantees distinct clear signatures in the voltage signal when a gesture is being performed above the photovoltaic surface. Figure 23 we can see only 11 gesture incidents out of a total of 150 were incorrectly classified.

We witness a similar trend in performance to our CNN model, where at a light intensity of 650 lux the right swipe gesture becomes the first gesture to become challenging to distinguish, until anticipating all right swipe gestures as double hand at 450 lux. Again, all other gestures operate flawlessly until we reach 350 lux, where we see a drop in precision (45%) and recall (40%), and more than half of our gestures are incorrectly detected. Random forest does not detect as many double hand gestures as CNN at 350 lux, but it does do somewhat better in identifying anticlockwise, clockwise, and left swipe gestures. Similarly, the model performed poorly at 250 lux, scoring an unacceptable accuracy of 26%.

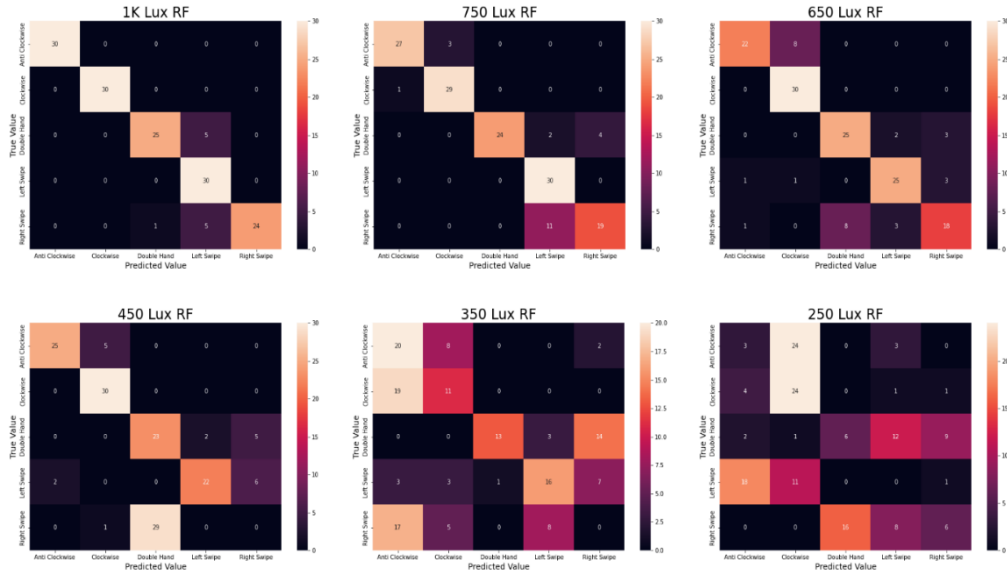


Figure 23 - Confusion Matrix for RF in various light intensities

Overall, our results suggest that both models are effective and are suitable for this application where environments are bright. Both models followed a similar trend in degradation when the light intensity is reduced, indicating that the poor accuracy is not attributable to the classification models, but to the ineffectual visible signatures left in the voltage signals produced by the photovoltaic sheet. Although the Random Forest model didn't perform as well as CNN in all light intensities, this was only by a margin. This can be seen in Figure 24.



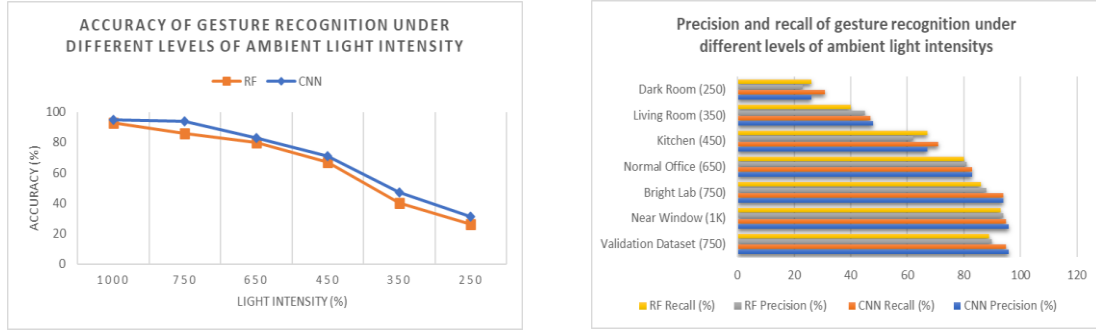


Figure 24 - Performance comparison of both CNN and RF over various light intensities

## 6.2 Experiment 2: The Flicker Effect

By executing the gestures beneath a flickering light, we can now explore the influence of sudden and abrupt changes in ambient light. Because we already know from experiment 1 that there is no performance deterioration at 750 lux, we tested our prototype in the same bright lab as to where the training dataset was collected for this study. To begin with, we have changed the light above the photovoltaic sheet to a flickering light, flashing at approximately 10Hz, oscillating between 750 lux to 550 lux at the photovoltaic sheet. The user performed 14 instances of each gesture below the flickering light (5 gestures x 14 repetitions = 70 total instances). Results show that both models are incapable of detecting the gestures when the light intensity is oscillating (31% accuracy for both RF and CNN). Changing the light intensities at such a rapid rate will degrade the voltage signal captured by the photovoltaic sheet, causing the voltage to leap up and down and deforming the signatures that the model has been trained to recognise. Although this scenario would be uncommon in any environment since the flashing light is noticeable to the naked eye.

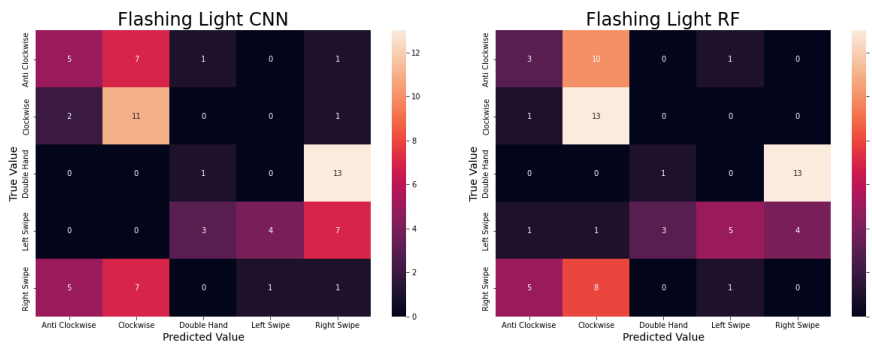


Figure 25 - Confusion Matrix of both CNN and RF when tested in Flashing Light

### 6.3 Experiment 3: The Importance of Principal Component Analysis

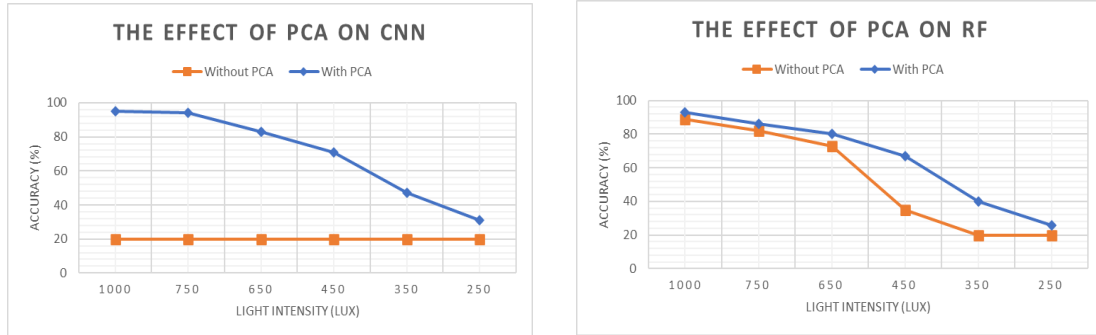


Figure 26 - The effect of PCA on both CNN and RF

Figure 26 shows the comparison between both models with and without the incorporation of Principal Component Analysis. A Convolutional Neural network with no PCA application shows no ability to learn. A straight line for accuracy denotes that the accuracy remains constant regardless of the intensity of the light. In fact, the CNN model was so inept that it identified all gestures as anti-clockwise for all light intensities. This could be owing to the capacity and complexity of our model. Our chosen CNN model was built for a small number of inputs, with only two convolutional layers: the first with the same number of neurons as the number of inputs, and the second with 400 neurons. When the CNN model is introduced with 12,000 raw data inputs instead of 15 principal components, the model tries to fit to the noise of the input data and so it becomes excessively specific, making it difficult to generalise to new training data. The small number of layers doesn't give the model a chance to learn all the intermediate features between the raw data and the high-level classification. When the model attempts to condense the feature representation from 12,000 features into 400 neurons found at the second convolutional layer, in a bottleneck fashion, it fails to sufficiently learn and preserve the most important features prior to classification.

With the same parameters as explored in Section 5.1, when PCA is applied to CNN the model can complete its training in 9 seconds, but CNN without-PCA took 57.2 seconds to complete and failed to learn. Increasing the number of neurons in the second convolutional layer to accommodate the increased number of inputs would only result in a significant increase in computation time and by a vast amount. When the number of nodes in the second convolutional layer was increased from 400 to 10,000 to accommodate the input size of CNN without-PCA, the model's accuracy on the validation subset climbed to only 31%, indicating that the model still requires a more complex larger architecture to learn. Although increasing the capacity of the model to this size increased the training time to 32 minutes and 26 seconds to complete. Therefore, it is clear to see that PCA showed an important role in both classification accuracy and computational time for CNN.

In Random Forest, however, we discover that the performance ability of a model with and without PCA is not significantly different. Although PCA increases the model's performance, particularly in low-light conditions, the Random Forest model can still learn just as well when fed vast amounts of raw data. Unlike our CNN model, Random forests are immune to feature magnitude, they are invariant to monotonic transformations of individual features. A random forest works by splitting a node based on a single feature, and the splitting of a feature is not influenced by other features. As a result, incorporating 12,000 features instead of 15 principal components has no discernible effect to the model's performance. What does enhance the model's ability to perform better is that PCA removes irrelevant, noisy, and redundant features from the feature vector, hence feeding the model with clean improved data will enhance the RF model's performance. With the same parameters explored in Section 5.2, when PCA is applied to RF the model took 9 seconds to complete its training, whereas a RF without PCA took 21 seconds to complete its training, hence demonstrating the importance of reducing the number of features the RF model must process.

Based on the test results of the two models with and without PCA, it can be concluded that the success of PCA can be used to improve the accuracy performance of both Convolutional Neural Network and Random Forest.

## 6.4 Experiment 4: The Importance of Normalisation

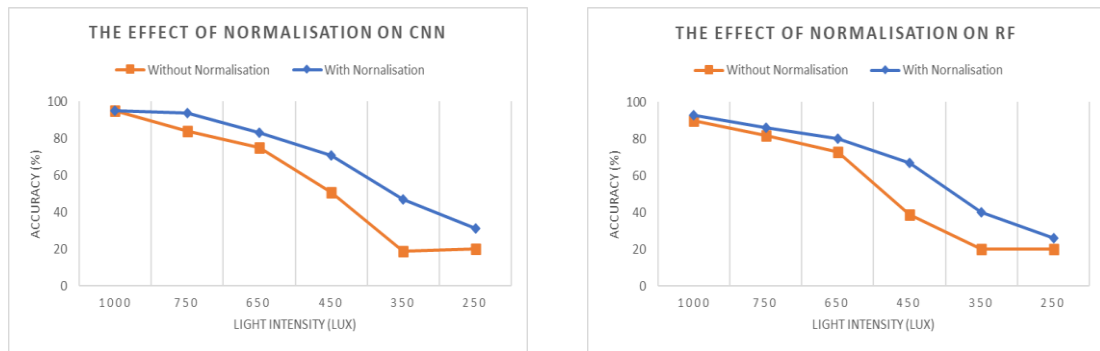


Figure 27 - The effect of Normalisation on both CNN and RF

Now that we've established the significance of PCA. In this experiment, we will evaluate the effectiveness of normalisation on the feature input. Figure 27 depicts the accuracy of both models when normalisation has been used and hasn't been used. The experimental results show that when normalisation is used, the accuracy is always higher than when normalisation has not been utilised. As mentioned in Section 4.3, we use normalisation to ensure that the magnitude values for all gestures made are scaled roughly the same regardless of the light intensity they were performed. As we can see, this pre-processing technique is capable of minimising the impact of light intensity on the model's performance to the greatest extent possible. As the intensity of the light declines, so does the magnitude of the voltage, and the signatures in the signal waveform become less and less visible. Both models show a sharper decline in performance without normalisation than when normalisation has been applied, particularly after 650 lux.

Based on the test results of the two models with and without normalisation, we can conclude that normalisation is essential for our final model due to its success in reducing the impact of both models' performances with light intensity.

Light Intensty (Lux)	Accuracy (%)					
	Without PCA nor Normalisation		With PCA, Without Normalisation		With PCA and Normalisation	
	CNN	RF	CNN	RF	CNN	RF
Near Window (1K)	20	89	95	90	95	93
Bright Lab (750)	20	82	84	82	94	86
Normal Office (650)	20	73	75	73	83	80
Kitchen (450)	20	35	51	39	71	67
Living Room (350)	20	20	19	20	47	40
Dark Room (250)	20	20	20	20	31	26

Table 3 - Final Table for all experiments 1, 3 and 4,



# Chapter 7

## Discussion

In this section, we will discuss the limitations of our study, the insight we have gained whilst conducting this work, and our future intentions to improve the research.

The proposed device requires bright light to function reliably. This device is particularly suitable in environments such as a bright lab or near a window during the day. Insufficient light poses as a limitation to the device, for example a living room at 350 lux. With conditions below 450 lux, the photovoltaic sheet fails to produce substantial signatures in the voltage signal which makes it challenging for the classifiers to distinguish the true meaning behind a command. Inconsistent light can also contribute to an unreliable performance of the proposed device. Currently, drastic variations in light, such as turning on and off a light source or the light emitted from a theatre screen will result in the voltage signal to be too noisy for the device to function reliably. In order to function, the proposed device requires sufficient light. In bright light, such as near a window at 1K, using CNN and RF, the system will perform flawlessly, achieving 95% and 93% accuracy, respectively.

To improve the representation of the raw voltage data, pre-processing techniques were applied. To capture the most significant features within the dataset, a dimensionality reduction technique was explored known as Principal Component Analysis (PCA). This technique works by projecting the most important features into a lower-dimensional subspace, eliminating noise and therefore simplifying the dataset without compromising the information extracted from the data. We were able to reduce the 12,000 voltage datapoints per gesture to 15 Principal Components. We discovered that 15 principal components are the bare minimum for retaining 99.007% of the variance in the dataset. This allowed us to reduce to input size for both models, cleaning the data from redundant extraneous features, improving both computation speed and performance accuracy.

The second pre-processing approach that we investigated was a scaling technique known as MinMax Normalisation. We discovered that environmental and human factors such as light intensity and hand proximity from the photovoltaic surface influenced the magnitude of the voltage produced by the photodiodes. For example, a gesture performed in a low light environment will produce small amplitude changes in the receiving voltage signal. Whereas a gesture performed in bright light intensity will produce substantially larger amplitude changes in the voltage signal. Mapping the features between 0 and 1, with 1 denoting the maximum magnitude and 0 denoting the minimum magnitude, ensured that all features were in the same range. This allowed the classifiers to compare distinct gestures made at varying light intensities with ease, maximising the performance of both models. With this strategy, we were able to reduce the impact of light intensity on the performance of both models to the greatest extent possible.

The gesture class used to train the proposed system was effective in bright light conditions. Each gesture class left a unique distinguishable signature in the amplitude of the voltage. Although the signatures of a gesture started to deteriorate as the light intensity was reduced. At 650 lux, a right swipe was the first gesture to become difficult to classify and was frequently misinterpreted as a double hand gesture. At 450 lux, a left swipe gesture also became harder to distinguish and was commonly mistaken as a right swipe gesture. Reducing to 350 lux, both anti-clockwise and clockwise signatures became increasingly difficult to distinguish between the two. The double hand gesture was the last gesture to become problematic, which could be due to the fact that it is the only gesture that does not require a direction.

Both clockwise/anticlockwise and left/right swipe gestures are inverted in time. With our current proposed system, we only have 1 dimension of voltage over a time series variable. This requires the user to always perform the clockwise/anticlockwise and left/right swipe gesture with the same hand that the model was trained with. To address this issue, we could propose that we divide the photovoltaic sheet into 4 electrical arrangements. One section to the right, one section to the left, another section at the top, and a final section at the bottom of the PV sheet. Doing so, having four independent voltage readings will provide the system with a sense of left/right and up/down direction when the user performs a gesture. This will minimise the confusion between inverted gestures and enable the system to recognise the same gesture regardless of which hand the user decides to perform it with.

All gestures used to train and test the proposed system were recorded by one user. The system is yet to be evaluated on other users. It will be impractical to expect all users to perform the same gesture in exactly the same way. Variations in user parameters such as hand size, hand angle, speed of the hand motion, the idea of how a gesture should be performed, and the proximity of the user's hand to the solar interface will all affect the signatures produced within the resultant voltage signal. The impacts of hand size, hand angle and proximity can all be reduced by the use of normalisation. Although, varying hand speeds amongst users will affect the classification of gestures as a result of the patterns within the signatures occurring at a different rate. This effect has yet to be mitigated. To compensate for time-temporal mismatches between multiple users, a possible solution is to apply dynamic time warping (DTW) to align the gestures during pre-processing. DTW is a time series analysis algorithm that has been used to compare comparisons between two temporal sequences that can differ in time and speed. With DTW, the system may be capable of reducing the impact of unavoidable human error parameters.

The system must be trained and evaluated using as many users as possible to make the system more robust. If the system doesn't work well on a large number of users, we could train and calibrate the system to a single user before use. Although, training of the system has been proven to be computationally expensive, this solution will require a decentralized approach. Because the proposed system is intended to be performed on a

micro-controller, to address this problem, a single user could connect the microcontroller to a standard computer via a serial connection during the training process. We could calibrate the system to a single user by requesting them to execute a series of gestures before use. The gestures performed during training will be transmitted through a serial connection, where the model will be trained on a standard computer with a python script. Once trained, the updated model can be uploaded back to the microcontroller as a serialised version ready to be deployed.

It is critical that our prediction models are both computationally inexpensive and demand little energy to function. In all lighting conditions, the CNN classifier outperformed the RF classifier in all metrics. However, we cannot dismiss the RF model just yet. The benefit of using a RF over the CNN is its simplicity to employ. Despite our efforts to keep the CNN architecture as minimal as possible, serialising the CNN model to transfer it to a microcontroller will result in a 35KB file size. A serialised trained RF, on the other hand, will be as little as 2.5 MB. Arduino microcontroller have become one of the most popular microcontrollers on the market and are very suited for this application. Taking a look at the Arduino Uno [43], Micro and Mega, they all offer a high frequency/clock speed at 16MHz. The EEPROM on the Uno and Micro are the same at 1 KB, while the Mega 2560 has 4 KB. All three storage capacities are acceptable for storing a serialised trained RF model, however the EEPROM capacity size is insufficient for a CNN classifier. Although, Micro SD card modules are available and can be attached to an Arduino to improve mass storage. This, however, may reduce computational speed.

Most importantly, the Arduino Uno, Micro and Mega microcontrollers support an ultra-low power consumption and can operate at a low voltage of 5V. An easy approach would be to trickle charge a rechargeable lithium battery or a supercapacitor, both of which would provide enough power on demand. A OPV3W60V DCDC converter can be used to convert the energy from the sun produced by the photovoltaic solar tape into a usable voltage to charge a lithium battery. The OPV circuit is a custom made DCDC converter for high-voltage photovoltaic devices with low power requirements. It will operate as a solar battery charger and safely charge an internal lithium polymer battery. The OPV circuit includes a Maximum Power Point Tracker (MPPC) to optimize the match between the solar array and the lithium polymer battery and to maximise the power generated by the photovoltaic sheet no matter the weather or indoor lighting conditions. Importantly, this will effectively charge the lithium battery quickly whilst not affecting the voltage signal used for classification. [44] [45]

Using the designed GUI, it can be noted that both models are capable of recognising a gesture in 0.05 seconds. Currently the GUI and photovoltaic sheet are separated due to the absence of a microcontroller and requires the user to browse on the GUI for a pre-recorded gesture. In the future the selected trained model will be deployed onto the microcontroller for gesture recognition, with its function to understand the meaning of the gesture before sending a command to the GUI. For the time being, found within the GUI script is the stored serialized object of both trained models for proof of work and demonstration. The GUI offers the ability to browse and select an unseen gesture from



the testing dataset. Once the unseen gesture has been selected, it only takes 0.05 seconds to load the gesture, pre-process the numeric data and predict the classification. As a result, I can confidently state that the system can operate in real time utilising these classifiers.

Currently, the results obtained from this project serve as a foundation for future development. In terms of gesture detection algorithms and gesture datasets, there is still a lot of space for improvement. However, our present work shows that using a large photovoltaic sheet for a gesture detection system in bright light conditions is a viable option. The project will be submitted towards the CHI Student Design Competition 2022. Both models and experimental tests can be found on GitHub:  
<https://github.com/georgequine/Masters-Code>

# Chapter 8

## Future Work

As discussed in Section 7, the current results obtained from this project serve as a foundation for future development. A list of potential ideas to improve the project will be presented below.

- To conduct a user study to investigate a wider dataset of gestures and to establish what action each gesture will accompany in an application.
- Additional experiments are required to determine the robustness of the system. Evaluations such as the direction of the light source to the photovoltaic sheet, the influence of bright saturated light (in direct sunlight, for example), and the effect of partial blockage of the system caused by clouds or nearby body movements. These would require further testing and pilot studies.
- Currently the suggested system is a single-user system. The system must be trained and tested on as many users as feasible in order to establish whether the model can be utilised globally for all users or whether it must be calibrated on an individual for a single use.
- If the system is to be calibrated for each user separately, it must be investigated how this may be accomplished effectively. Training of the system has been proven to be computationally expensive. Therefore, currently the training process is performed on a standard computer.
- To investigate the proposed idea for dividing the photovoltaic sheet into four independent sections to provide the system with a sense of left/right and up/down direction, minimising the confusion between inverted gestures and allowing the system to recognise the same gesture regardless of which hand the user chooses to perform it with.
- To investigate the effectiveness of Dynamic Time Warping (DTW) to compensate for time-temporal mismatches between multiple users when performing a gesture at varying hand speeds.
- We do not want the microcontroller to be continuously detecting for gestures when the system is not in use. This will consume unnecessary computational power. Therefore, we will need to investigate a calibration step to detect the start and end of a gesture. Perhaps a threshold-based approach could be adopted. When the microcontroller detects a voltage drop below the threshold, the microcontroller will begin to collect a time-series result. Once the voltage returns above the threshold, the microcontroller will consider the gesture to be over and stop recording.
- We will look into the minimum sampling rate that the system can effectively operate with. Currently, the voltage will be read at a sample rate of 3,000 times per second. Although, the higher the sampling rate, the more processing power is required by the microcontroller. It is of the highest priorities to reduce the

computational overhead. To accomplish this, we will down sample the initial sampling speed in even intervals of 250 until a minimum of 250 samples per second. The classifier will then be retrained for each sampling rate, and the system's accuracy will be assessed.

- Currently the system does not work well in noisy backgrounds with abrupt changes in lighting. If the system is to be used within a busy store to control a screen of display, for example, it is critical that background user movement does not interfere with the user's ability to use the system. As a result, we will look at a smoothing technique to reduce big amplitude shifts caused by noisy backgrounds.
- The microcontroller and the photovoltaic sheet must all be implemented for final deployment. To avoid hazards, the circuit should be hidden from the end user. With the use of a spring, I propose the photovoltaic sheet can be withdrawn from a tightly wrapped cylindrical casing before application use and then neatly sprung back once the user has finished with the product. This will provide an ideal solution for portable applications.
- Once the entire circuit has been constructed, we must assess the power needs required by the system during its active operation of gesture recognition. We will need to determine whether the harvested energy exceeds the systems needs in a series of various lighting conditions.
- We must determine if the suggested system is better suited to a LiPo battery that is continuously charged via the DCDC converter or the use of a supercapacitor.
- To implement the system to operate current third-party applications that exist today such a Smart TV, sliding through a catalogue and selecting an option by gestures being performed above the photovoltaic sheet.

# Chapter 9

## Conclusion

In this paper, we proposed a self-powered rollable gestural interface using ambient light for both energy harvesting and gesture recognition. We developed a model that can detect a hand gesture conducted above the photodiodes in bright light conditions in less than 0.05 seconds.

There are numerous applications in this discipline, including virtual reality, sign language, and smart robotic control. When compared to other gesture recognition approaches, employing ambient light to identify gestures have various advantages. For starters, there are no privacy concerns that camera-based systems encounter. There is no need to invade the user's privacy by obtaining the image input from a camera in order to gain information of the user's intentions. It is virtually impossible to maliciously create anything from the data of a silhouette that our proposed approach receives. In comparison to contact-based devices, the system may be used totally hands-free, providing the user with greater comfort. The energy harvesting capabilities of our proposed system also offer a perfect solution for portable use, eliminating the inconvenience of replacing or recharging batteries as well as the difficulty of finding a nearby power resource. Additionally, the flexible rollable material enables the system to be deployed over undulating surfaces where its display size and form factor can be dynamically changed. This makes our system useful for consumers where space, size and surface characteristics vary.

We observed that each hand gesture produced a unique shadow pattern over the photovoltaic sheet. This left distinguishable signatures in the amplitude of the harvested voltage. The voltage signal was recorded for each gesture and stored as a CSV file in folders that were named according to the gesture executed. Pre-processing techniques were applied to the raw voltage data to reduce inherent noise caused by inevitable environmental and human factors. Normalisation and Principal Component Analysis both increased the proposed system's performance in all tested lighting conditions. The resultant data was shuffled and randomly split into two separate datasets, a training dataset and a validation dataset. To determine the true meaning underlying the executed gestures, both machine and deep learning techniques were applied. We focused on two classifiers, one utilising a machine learning technique, Random Forest, and the other a deep learning classifier, 1D -Convolutional Neural Network.

After careful tuning of the hyperparameters using a Weights & Biases python tool, we evaluated the performance of the system. We assessed both models' robustness across six unseen different light intensities, including a near window (1K lux), a bright lab (750 lux), a normal office (650 lux), a kitchen (450 lux), a living room (350 lux), and a dark room (250 lux). The CNN showed the best overall accuracy in all lighting conditions, achieving a maximum of 95% accuracy in 1K lux. The RF performed equally as well, achieving 93% accuracy in 1K lux. It is apparent that when the light intensity is reduced, the photovoltaic sheet struggled to produce substantial signatures in the voltage signal which made it challenging for the classifiers to distinguish the true meaning behind a given command.

With this work and the anticipated future work in mind, we successfully demonstrated the viability of using a large photovoltaic sheet for both energy harvesting and gesture recognition using bright light. In today's society, the emphasis must be on sustainability and environmental protection. This self-powered rollable gesture interface will be beneficial in situations where resources are restricted but light is available (sun). In the future, I hope that our project will motivate academics and designers to create new unique use cases and applications based on self-powered devices using photovoltaic sheets.

# Chapter 10

## Bibliography

- [1] Richard Harper, Tom Rodden, Yvonne Rogers, Abigail Sellen. Human-Computer Interaction In the Year 2020. Microsoft Research Ltd. 2008. Retrieved from: [https://blogs.commonsg.georgetown.edu/ccp-797-fall2013/files/2013/12/beinghuman\\_a3.pdf](https://blogs.commonsg.georgetown.edu/ccp-797-fall2013/files/2013/12/beinghuman_a3.pdf)
- [2] Fitbit Inc. Fitbit. Available at: <https://www.fitbit.com/>.
- [3] Apple Inc. Apple iPad. Available at: <http://www.apple.com/ipad/>
- [4] June Life Inc. June oven. Available at: <https://juneoven.com/>
- [5] Joshua r. New, Erion Hasanbelliu, Mario Aguilar. Facilitating User Interaction with Complex Systems via Hand Gesture Recognition. 2003. Retrieved from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.132.49>
- [6] Faheem Khan, Seong Kyu Leem, Sung Ho Cho. Hand-Based Gesture Recognition for Vehicular Applications Using IR-UWB Radar. 2017. Retrieved from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5422194/>
- [7] Microsoft Corp. Xbox. Kinect. Available at: <http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one>
- [8] Leap Motion. Available at: <https://www.leapmotion.com/>
- [9] Nintendo Wii. Nintendo. Available at: <https://www.nintendo.co.uk/Wii/Accessories/Accessories-Wii-Nintendo-UK-626430.html>
- [10] CyberGlove III. Cyber Glove Systems. Available at: <http://www.cyberglovesystems.com/cyberglove-iii>
- [11] G. D. Kessler, L. F. Hodges, and N. Walker. Evaluation of the cyberglove as a whole-hand input device. 1995. Retrieved from: [https://www.researchgate.net/publication/27521573\\_Evaluation\\_of\\_the\\_CyberGloveTM\\_as\\_a\\_Whole\\_Hand\\_Input\\_Device](https://www.researchgate.net/publication/27521573_Evaluation_of_the_CyberGloveTM_as_a_Whole_Hand_Input_Device)
- [12] Vamsi Talla, Bryce Kellogg, Shyam Gollakota, Joshua R Smith. Battery-Free Cellphone. 2017. Retrieved from: <https://dl.acm.org/doi/10.1145/3090090>
- [13] Vinod Kumar, R.L. Shrivastava, S.P. Untawale. Solar Energy: Review of Potential Green & Clean Energy for Coastal and Offshore Applications. 2015. Retrieved from: <https://www.sciencedirect.com/science/article/pii/S2214241X15000632>
- [14] Arun Kumar, S.K. Shukla. A Review on Thermal Energy Storage Unit for Solar Thermal Power Plant Application. 2015. Retrieved from: <https://www.sciencedirect.com/science/article/pii/S1876610215014964>
- [15] Le-Giang Tran, Hyouk-Kyu Cha, Woo-Tae Park. RF power harvesting: a review on designing methodologies and applications. 2017. Retrieved from: <https://mmsl-journal.springeropen.com/articles/10.1186/s40486-017-0051-0>.
- [16] Radka Pavelkova, David Vala, Katerina Gecova. Energy harvesting systems using human body motion. 2018. Retrieved from: <https://www.sciencedirect.com/science/article/pii/S240589631830870>
- [17] Michella Ma. University of Washington. Battery-free technology brings gesture recognition to add devices. 2014. Retrieved from: <https://www.washington.edu/news/2014/02/27/battery-free-technology-brings-gesture-recognition-to-all-devices/>
- [18] S.P. Beeby, Z. Cao, A. Almussallam. Kinetic, thermoelectric and solar energy harvesting technologies for smart textiles. 2013. Retrieved from: <https://www.sciencedirect.com/science/article/pii/B978085709342450110>
- [19] Oculus. Virtual Reality. Available from: <https://www.oculus.com/>
- [20] Iker Vasquez. Boise State University. Hand Gesture Recognition for Sign Language Transcription. 2017. Retrieved from: <https://scholarworks.boisestate.edu/cgi/viewcontent.cgi?article=2322&context=td>
- [21] Felix Cruetzig, Peter Agoston, Jan Christoph Goldschmidt. The underestimated potential of solar energy to mitigate climate change. 2017. Retrieved from: [https://www.researchgate.net/publication/319396014\\_The\\_underestimated\\_potential\\_of\\_solar\\_energy\\_to\\_mitigate\\_climate\\_change](https://www.researchgate.net/publication/319396014_The_underestimated_potential_of_solar_energy_to_mitigate_climate_change)
- [22] Built in 2019. What is Artificial Intelligence? How Does AI Work? Retrieved from: <https://builtin.com/artificial-intelligence>
- [23] Christian Sorescu, Yogesh Kumar Meena, Deepak Ranjan Sahoo. PViMat: A Self-Powered Portable and Rollable Large Area Gestural Interface Using Indoor Light. 2020. Retrieved from: <https://dl.acm.org/doi/10.1145/3379350.3416192>
- [24] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, Mushfika B. Upama, Ashrad Uddin, Mostafa Youssef. SolarGest: Ubiquitous and Battery-free Gesture Recognition using Solar Cells. 2019. Retrieved from: <https://doi.org/10.1145/3300061.3300129>
- [25] Mahina-Diana A. Kaholokula. Reusing Ambient Light to Recognize Hand Gestures. 2016. Retrieved from: [https://digitalcommons.dartmouth.edu/cgi/viewcontent.cgi?article=1104&context=senior\\_theses](https://digitalcommons.dartmouth.edu/cgi/viewcontent.cgi?article=1104&context=senior_theses)
- [26] Heba Abdelnasser, Moustafa Youssef, Khaled A. Harras. WiGest: A Ubiquitous WiFi-based Gesture Recognition System. 2015. Retrieved from:

<https://arxiv.org/pdf/1501.04301.pdf>

- [27] Yichen Li, Tianxing Li, Ruchir A. Patel, Xing-Dong Yang, Xia Zhou. Self-Powered Gesture Recognition with Ambient Light. 2018. Retrieved from: <http://dx.doi.org/10.1145/3242587.324263>
- [28] infinityPV Solar Tape. OPV. Available from: <https://infinitypv.com/products/opv/solar-tape>
- [29] infinityPV. OPV brochure. Available from: [https://infinitypv.com/images/infinityPV\\_OPV\\_organic\\_solar\\_cells.pdf](https://infinitypv.com/images/infinityPV_OPV_organic_solar_cells.pdf)
- [30] PicoScope. PicoScope 5000 Series. Available from: <https://www.picotech.com/oscilloscope/5000/flexible-resolution-oscilloscope>
- [31] Numpy. Available from: <https://numpy.org/>
- [32] Scikit-learn. Available from: <https://scikit-learn.org/stable/>
- [33] Google Colab. Available from: [https://colab.research.google.com/?utm\\_source=scs-index](https://colab.research.google.com/?utm_source=scs-index)
- [34] Python. Tkinter Interface. Available from: <https://docs.python.org/3/library/tkinter.html>
- [35] Sartorius. What Is Principal Component Analysis (PCA) and How It Is Used? 2021. Available from: <https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186>
- [36] Towards Data Science. A One-Stop Shop for Principal Component Analysis. 2021. Available from: <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>
- [37] Tony You. Towards Data Science. Understanding Random Forest. 2019. Available from: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
- [38] Niklas Donges. BuiltIn. A complete Guide to the Random Forest Algorithm. 2020. Available from: <https://builtin.com/data-science/random-forest-algorithm>
- [39] Sarah Deweerdt. Deep connections. 2019. Available from: <https://media.nature.com/original/magazine-assets/d41586-019-02208-0/d41586-019-02208-0.pdf>
- [40] Nils. Good Audience. Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences. 2018. Available from: <https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>
- [41] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman. 1D convolutional neural networks and applications: A survey. 2021. Available From: <https://www.sciencedirect.com/science/article/pii/S0888327020307846>
- [42] Yuanhao Xiong, Yan Liu, Xu Sun. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. 2019. Available From: <https://www.luolc.com/publications/adabound/>
- [43] Arduino. Arduino Nano 33 Ble. Available from: <https://store.arduino.cc/arduino-nano-33-ble>
- [44] OPV3W60V – DCDC Converter – OPV. Available from: <https://infinitypv.com/products/electronics/opv3w60v>
- [45] OPV3W60V MPPC. Application Notes. OPV. Available from: [https://infinitypv.com/application\\_notes/OPV3W60V\\_applicationnoteV1.1.pdf](https://infinitypv.com/application_notes/OPV3W60V_applicationnoteV1.1.pdf)