OBJECT RECOGNITION USING SOFT SENSORS

LIM CHU CHEN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Electrical and Electronics Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

Jan 2021

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _Chen_

Name : Lim Chu Chen
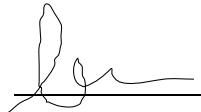
ID No. : 1602697

Date : 19/4/2021

I certify that this project report entitled **"OBJECT RECOGNITION USING SOFT SENSOR"** was prepared by **LIM CHU CHEN** has met the required standard for submission in partial fulfilment of the requirements for the award Bachelor of Engineering (Honours) Electrical and Electronics Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____
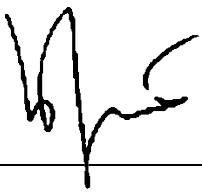
Supervisor : Dr. Chee Pei Song

Date : 19/4/2021

Signature : _____

Co-Supervisor : Prof. Ts. Dr. Lim Eng Hock

Date : 19/4/2021

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank everyone who has contributed to the successful completion of this project. I would like to express my sincere gratitude to my research supervisor, Dr. Chee Pei Song for his invaluable advices and guidance throughout the development of this project.

On top of that, I would also like to thank my co-supervisor, Prof. Ts. Dr. Lim Eng Hock for his guidance and support as well. In addition, I would like to express my gratitude to my loving parents, and my friends who had helped and given me encouragement throughout the entire process of writing this paper.

ABSTRACT

The development of soft sensors with high sensitivities and good response time is currently researched in great interest, especially in healthcare and soft robotics systems. However, there is a lack of study to equip the soft sensors with a smart feature. Therefore, this study proposes a smart glove that can recognise objects using a support vector machine (SVM), a supervised machine learning algorithm. The input to the smart glove is obtained from the integrated resistive strain-based flexible sensors. The characterisation of the resistive sensor was done, and the sensitivity was found to be 0.0145 kΩ/°. The glove is able to recognise three distinct object shapes with an accuracy of up to 92%. Through AI-based object recognition and its high accuracy, this glove provides a promising solution for a low-cost soft sensor solution for the area of soft robotics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

R   Resistance, Ω

P   Pressure, Pa

T   Temperature, $°C$

ρ Density, kg/m^3

v Poisson's ratio

ϵ_r Relative permittivity

E   Young's Modulus, P_a

eGaIn   Eutectic Gallium-Indium alloy

IPMC   Ionic Polymer Metal Composite

CS3   Capacitive Soft Strain Sensor

FEA   Finite Element Analysis

NaCl   Sodium Chloride

TPE   Thermoplastic Elastomer

CAD   Computer Aided Drawing

SVM   Support Vector Machines

CNN   Convolutional Neural Network

LiDAR   Light Detection and Ranging

CHAPTER 1

INTRODUCTION

1.1   Background

Soft robotic manipulators such as soft grippers are a growing area of research. A fully developed soft gripper allows for increased adaptability and compliance for the gripper in less well-defined environments in which the traditional rigid robotics are not suitable (Kier & Stella, 2007). This allows for higher flexibility in the area of manufacturing or logistics.

However, with a less defined environment for the soft gripper to work with, there has to be a continual feedback system for the accurate control of the gripper and obtaining real-time information from its environment. This poses a significant current challenge as soft systems do not have a limited and constrained degree of freedom (Hughes et al., 2016). Thus, soft sensors are usually integrated within the gripper to ensure a continual feedback system.

With the world shifting towards the brink of the Fourth Industrial Revolution, the field of predictive modelling and artificial intelligence has advanced to the point that it can be applied to almost all situations with relative ease. In addition, with the digitisation of data and open-source licensed training models, the data needed to train a neural network can be easily obtained online with minimal cost. Thus, with the benefits that a predictive model can provide for a system, there would be no reason not to integrate predictive modelling with soft sensors for object recognition.

Therefore, this report discusses a potential feedback system using a resistive based approach for the applications that can be applied in several industries, including the soft robotics and healthcare industry. On top of that, a machine learning algorithm will be implemented for object recognition.

1.2   Problem Statement

In the next few years, an increased number of broadly-connected devices are expected to be deployed; the main trend towards automation and the Internet of Things era. More and more technological devices are expected to be automated with minimal human input. These new inter-connected devices will be the evolution of traditional existing devices and will ease with the sharing of data between devices and processing them. However, an increased amount of automation will consequently produce a large set of data. The data obtained has to be processed are crucial to improving an automated system. Traditionally, large amounts of data are processed by statistics and manually finding patterns within the data set and making a generalisation for the data which is time and resource intensive. Therefore, a solution would be to use a self-learning algorithm which automatically finds generalisations based on its inputs and has the potential to be improved with more data, also known as artificial intelligence or machine learning.

1.3   Aims and Objectives

The main aim of this project was to fabricate a prototype glove using soft sensors that has the ability for object recognition based on the resistive strain mechanism; with the sensor being used to detect flexing and angle changes of the finger.

The following objectives were set to achieve the aim:

- To characterise resistive based sensor at different bending angles.

- To fabricate a smart glove prototype based on a resistive strain principle.

- To integrate an AI algorithm into the smart glove to recognise the shape of an object.

1.4 Report Overview

In the following chapter, the overall literature review of existing flexible sensor solutions, the pros and cons of existing solutions, and the existing design and parameters used for the sensors were discussed. The different AI methods that can be used for object recognition and the pros and cons were also understood and studied. Finally, the different types of kirigami structure and ways of improving the stress-strain relationship was explained.

In chapter 3, the methodology of this study, the FEA simulation and the structural design of the kirigami structure were explained. On top of that, the characterisation of the resistive flex sensor and the training and modelling steps of the machine learning algorithm was done and visualised in this chapter. The results were then presented in chapter 4 of this paper.

Finally, a conclusion was drawn in chapter 5, with the recommendations and suggested future works about different ways that the current prototype can be achieved.
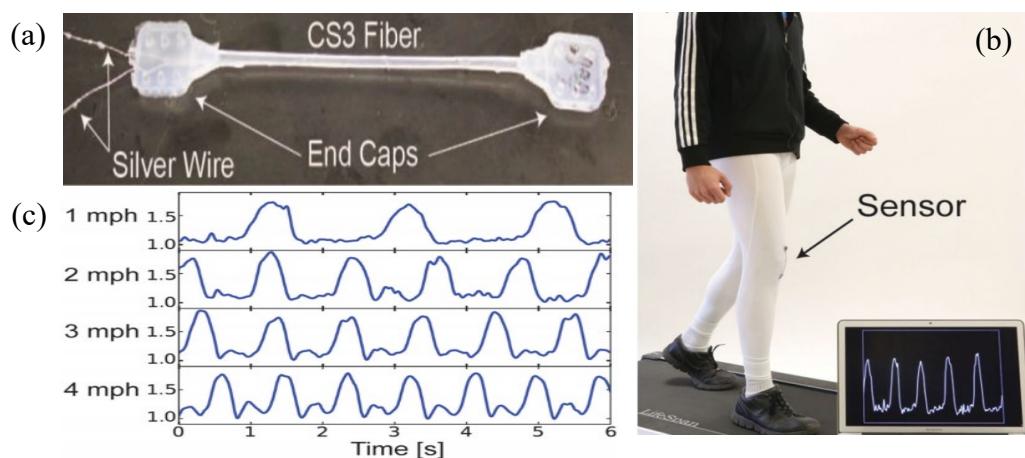
CHAPTER 2

LITERATURE REVIEW

2.1   Methods of Transduction

2.1.1   Capacitive Strain Sensors
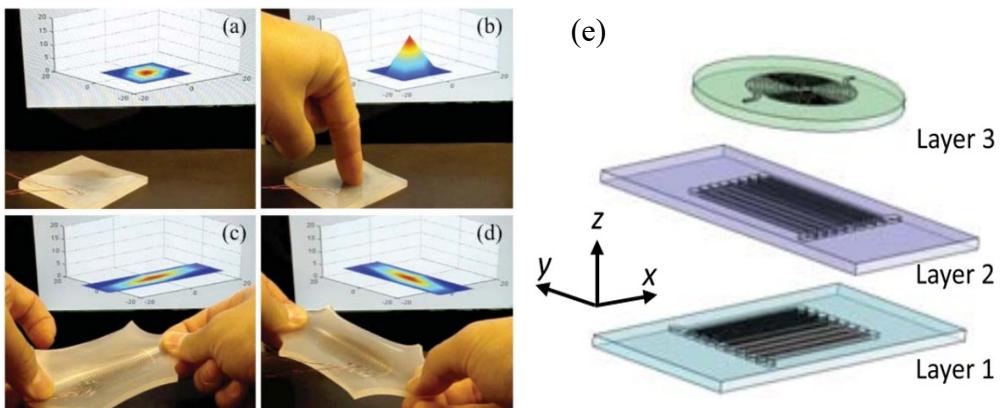
Capacitive soft strain sensors are used to detect elongation strains. Capacitive sensors usually consist of two conductive layers with a dielectric layer in between. The thickness changes in response to an applied deformation. With a change in the thickness, the resultant capacitance of the strain sensor changes (Frutiger et al., 2015). These sensors have the advantage of minimal hysteresis, which usually occurs due to changes in electrical resistance due to high strain situations. Carbon nanotubes capacitive strain gauges are able to measure strain as large as 300% with high sensitivity (Cai et al., 2013). However, due to their difficult manufacturing processes, they are not as economical to produce as the other types. They are difficult to apply in the case of object recognition, as the morphology of the sensors is limited. Figure 2.1(a) shows a capacitive soft strain sensor (CS3) with silver wires sewn through the end caps to inhibit connection loss and the signal produced (Figure 2.1(c)) while the subject is walking at a different pace when the sensor is attached as shown in Figure 2.1(b).

Figure 2.1:Capacitive Sensor (a) Moulded Capacitive Sensor with Protruding Silver Wires. (b) Sensor Mounted on a Textile Attached Across the Knee. (c) Signal Measured with Different Walking Speed (Frutiger et al., 2015)

2.1.2   Soft Strain Ionic Sensors
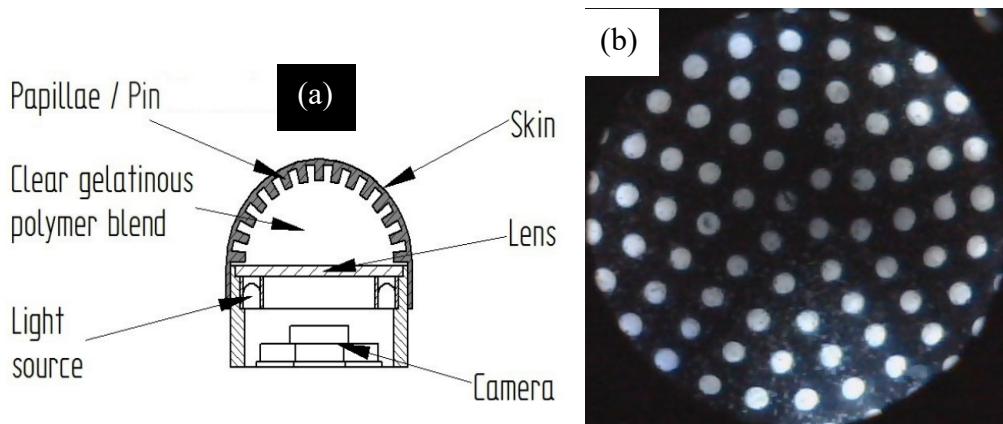
Soft strain ionic sensors are silicon-based resistive strain gauges that can achieve more than 10% flexibility(Keulemans et al., 2014, Park et al., 2012). These sensors are composed of an elastomer which has channels within the material, and the channels are filled with electrically conductive liquid such as an eutectic gallium-indium (EGaIn) alloy, as shown in Figure 2.2(a-d) (Park et al., 2012) or sodium chloride(NaCl) solution (Cheung et al., 2008). When force is applied axially, the thin elastomer deforms and the liquid within the elastomer changes in length and cross-sectional area, resulting in minute changes in resistance. With this characteristic, multiple sensors can be layered on top of each other to form 3-dimensional sensing, as shown in Figure 2.2(e). Due to their linearity of output over a wide range of strain, ionic strain gauges are a good candidate for soft robotics applications (Chossat et al., 2013). These sensors must be carefully and accurately designed to ensure the ionic channels measures a particular degree of freedom(Hughes et al., 2016).

Figure 2.2: Soft Strain Ionic Sensor (a) No stimulus. (b) Contact Pressure. (c) x-axis strain. (d) y-axis strain. (e) Three sensors layered to detect x, y and z strains. (Park et al., 2012)

2.1.3   Optical Sensors
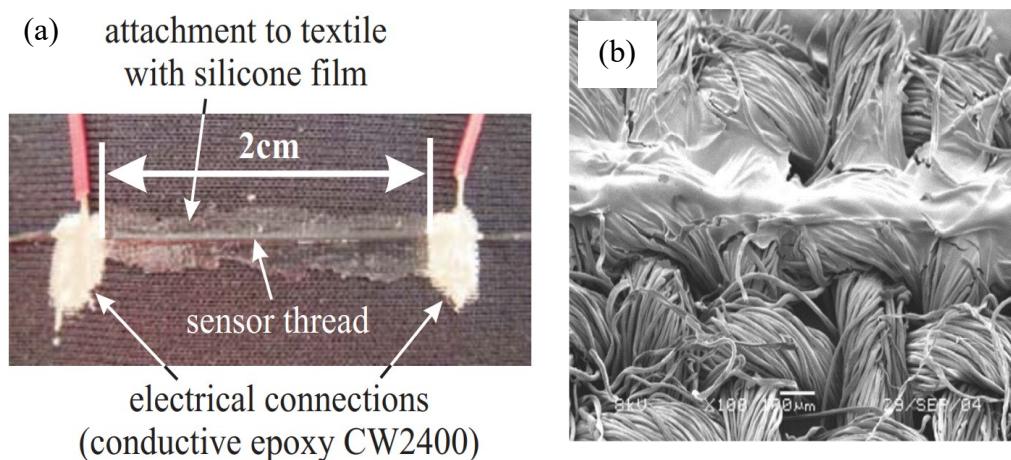
This sensor contacts objects with a moulded silicon rubber dome filled with clear silicone gel. The inside of the tip comprises of a series of geometrically arranged white-tipped pins as shown in Figure 2.3(a-b) (Winstone et al., 2013). Deformation from device-object interaction is then measured with a camera by tracking the movement of the white-tipped pins inside the rubber dome. Optical sensors work based on the principles of deformation of the epidermal layers of the human skin and is able to measure greater displacements compared to conventional sensors(Chossat et al., 2013). However, optical sensors are not suitable for soft robotics due to their bulkiness and difficulty of integration into soft structures (Chossat et al., 2013). Other alternatives of optical sensors include a differential optical-fiber displacement sensor which uses fiber optics and a laser diode light source to estimate a measurement (Suganuma et al., 1999).

Figure 2.3:Optical sensor   (a)Cross section of an optical-based sensor. (b)raw image from the camera inside the sensor showing white tipped pins (Winstone et al., 2013).

2.1.4   Strain Sensitive Fibers
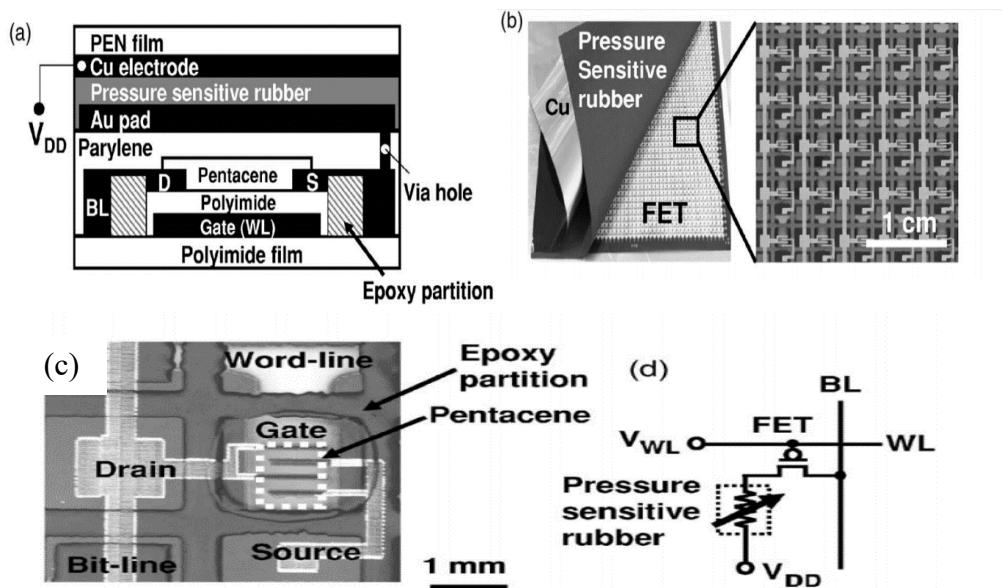
Strain fibers and textiles have been developed to sense user activities with many applications, such as in physiotherapy and sports. These sensors are attached to textiles such as clothing to be able to detect body postures with a measurement range of 100% strain, as shown in Figure 2.4(a) (Tognetti et al., 2005). These sensors are fiber-shaped and consist of a thermoplastic elastomer (TPE) filled with carbon powder which changes its resistivity with length (Mattmann et al., 2007). There are also other types that are printed directly onto fabric (Calvert et al., 2008). One disadvantage of using these sensors is that the strains measured by these sensors are significantly lower than measured ionic strain sensors (Hughes et al., 2016). Figure 2.4(b) shows a conductive sensor printed on cotton with PEDOT-PPS material (Calvert et al., 2008).

Figure 2.4: Strain Sensitive Fibers (a) Sensors attached to fabric (Mattmann et al., 2007). (b) Conductive Sensor printed on cotton (Calvert et al., 2008).

2.1.5   Flexible Electronics
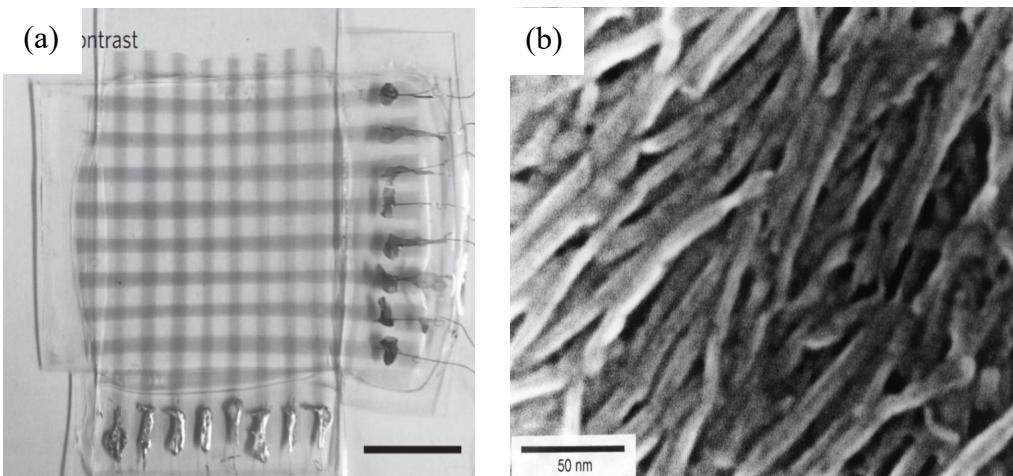
Field-effect transistors (FETs) have been increasingly used in flexible electronics applications due to their size. FETs combined with rugged, lightweight plastics and elastomers can provide a low-cost, high mechanical flexibility electrical circuit.(Rogers et al., 2001). These applications are widely studied in the field of flexible display but can create a high-performance pressure sensor by arranging pentacene FETs in an active matrix layer. These sensors are made with polyimide precursors and silver nanoparticles patterned on a polyimide film using an ink-jet printing system (Noguchi et al., 2006). Figure 2.5(a) and 2.5(c) show an overview of the working mechanisms that form a FET pressure sensor, and Figures 2.5(b) shows the active matrix of the mechanism. Figure 2.5(d) shows the schematic diagram of a single pressure-sensitive FET module.

Figure 2.5: Flexible Electronics (a) Cross section of pressure sensor (b) Pressure sensing mat consisting of organic FET active matrix. (c) One cell of pentacene FETs. (d) circuit diagram of a stand-alone FET cell.

2.1.6 Carbon-nanotube Resistive Strain Sensor
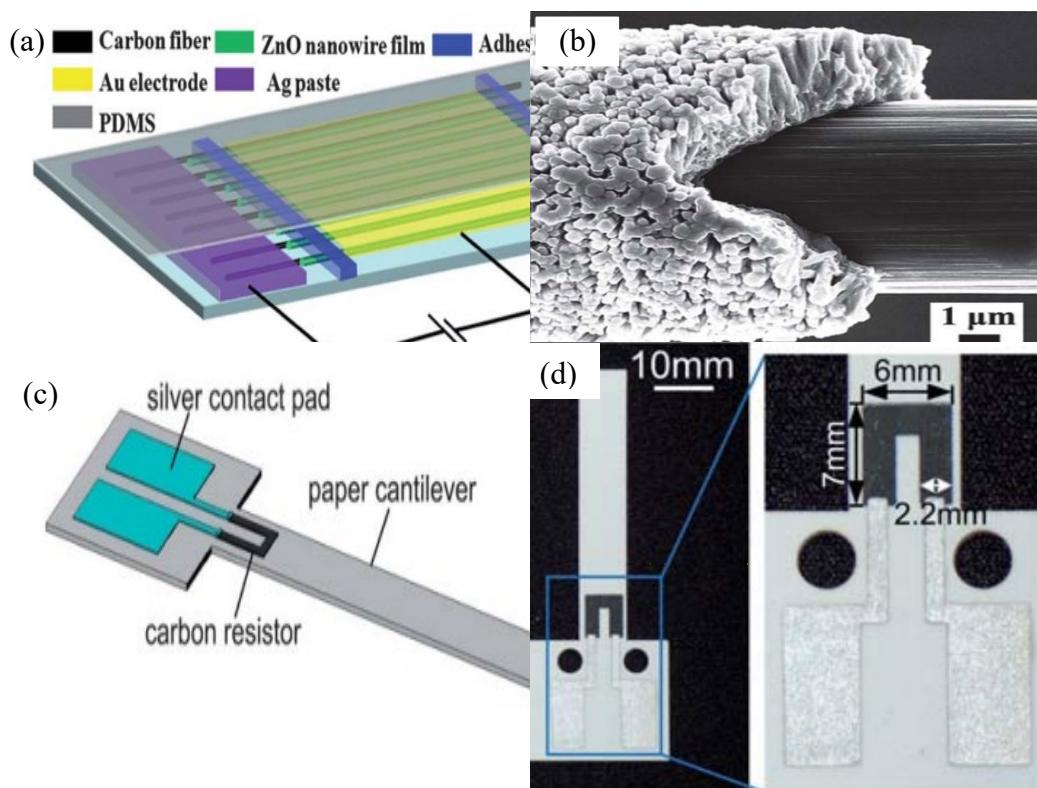
Carbon is often formed into a matrix and integrated into thermoplastics and elastomers such as silicone to produce highly flexible sensors. For example, Carbon-nanotubes are used to provide 'skin-like' sensors, but there has been limited integration of these sensors in the robotics application (Lipomi et al., 2011). Due to the high density and the minute size of the carbon nanotubes, a more conductive network within the sensor(Shaffer et al., 1998) may increase the efficiency of these sensors. However, these sensor types might be easily influenced by mechanical disturbances, like stress or shear(Pham et al., 2016). Figure 2.6(a) shows a stitched carbon nanotube strain sensor. Figure 2.6(b) shows the nanotube lines in an electron microscope.

Figure 2.6: Carbon Nanotube Resistive Strain (a) Lines of nanotube shown in a high contrast photograph (Lipomi et al., 2011). (b) Scanning Electron Microscope (SEM) of the surface of a nanotube film (Shaffer et al., 1998).

2.1.7   Piezoeletric Strain Sensor

Several types of piezoelectric sensors have been developed using nanotechnologies such as using a single ZnO piezoelectric fine wire (Zhou et al.,2008) and hybrid carbon fiber structures and ZnO (Liao et al., 2013). The I-V characteristics scale linearly and are highly sensitive to strain (Zhou et al., 2008), but the range of strains these sensors can handle is low(Hughes et al., 2016). Flexible and cheap paper-based piezoresistive sensors were also developed which uses MEMs technology (Liu et al., 2011).

Figure 2.7:Piezoelectric Strain Sensor (a)Schematic diagram of a carbon fiber-ZnO hybrid structure (Liao et al., 2013). (b) Cutaway view SEM image of the hybrid structure (Liao et al., 2013). (c) Schematic diagram of a paper-based piezoresistive sensor (Liu et al.,2011)

2.1.8   Triboelectric Generators
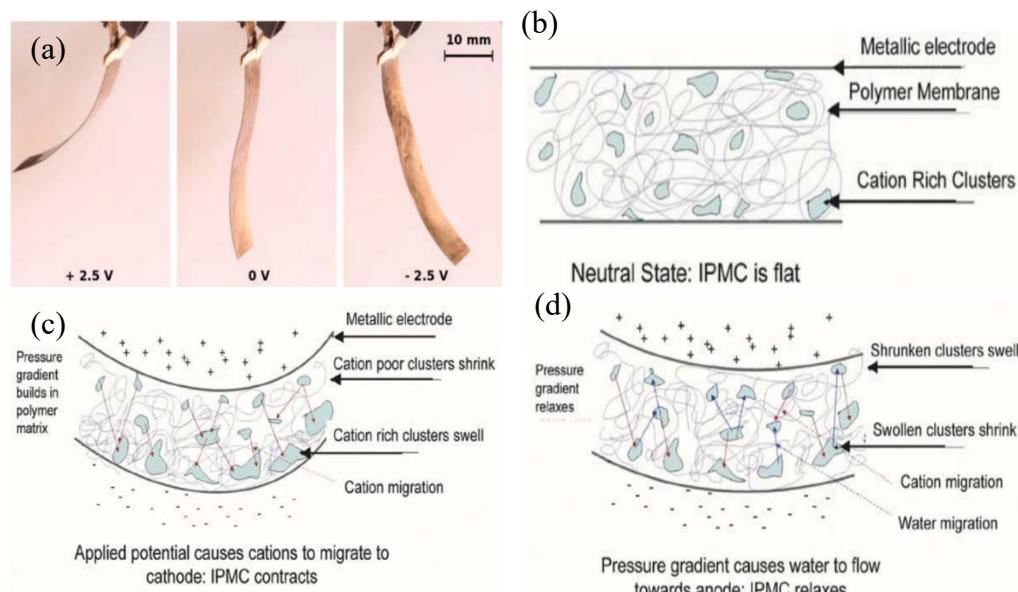
The triboelectric effect or contact-electrification is where some materials can store charge after contacting with a different material. Numerous studies on the workings of this phenomena include theories such as the correlation between the charge amount with bandgaps, the ion densities, and work functions (Xu et al., 2018). Usually, charges induced in this process is considered wasted energy (Fan et al., 2012). However, triboelectric nano generators were discovered back in 2012 (Wang., 2012), and further development on the technology was done to create a flexible power source (Wang et al., 2017). A triboelectric generator (TEG) consists of multiple different polymers stacked alternatively with a thin layer (100nm) of Au alloy film-coated to the top and bottom layer (Fan et al., 2012). TEGs can produce an electrical potential as the material separates (Wang et al., 2017).

Figure 2.7: (a) The structure of a TEG in bending and releasing process and the related electrical measurements (Fan et al., 2012).(b) Theoretical model of dielectric sandwiching (Wang et al., 2017). (c) Proposed mechanism of a TEG (Fan et al., 2012). (d)TEG Prototype (Fan et al., 2012).

2.1.9   Ionic Polymer Metal Composite

Ionic Polymer Metal Composite (IPMC) is a synthetic material that can show large deformations when an electric field is applied that behaves similarly to biological muscles for biomechanics operations as shown in Figure 2.8(a)(Shahinpoor et al., 1998). Conversely, when the electroactive polymer is mechanically bent or deformed, an output voltage can be measured based on the Poisson-Nernst-Planck field theories (Pugal et al., 2011). IPMC usually consists of two metal electrodes and free cations dissolved in pockets of solvent (e.g. water) with polymer membranes sandwiched within, as shown in Figure 2.8(b). When a voltage is applied, cations travel to the cathode electrode, followed by the water. This movement cause a hydrophilic expansion which results in the bending of the polymer, as shown in Figure 2.8(c & d) (Madden et al., 2004). IPMCs have advantages such as lightweight, good flexibility, silent operation, and low actuation voltage (Chen., 2019).
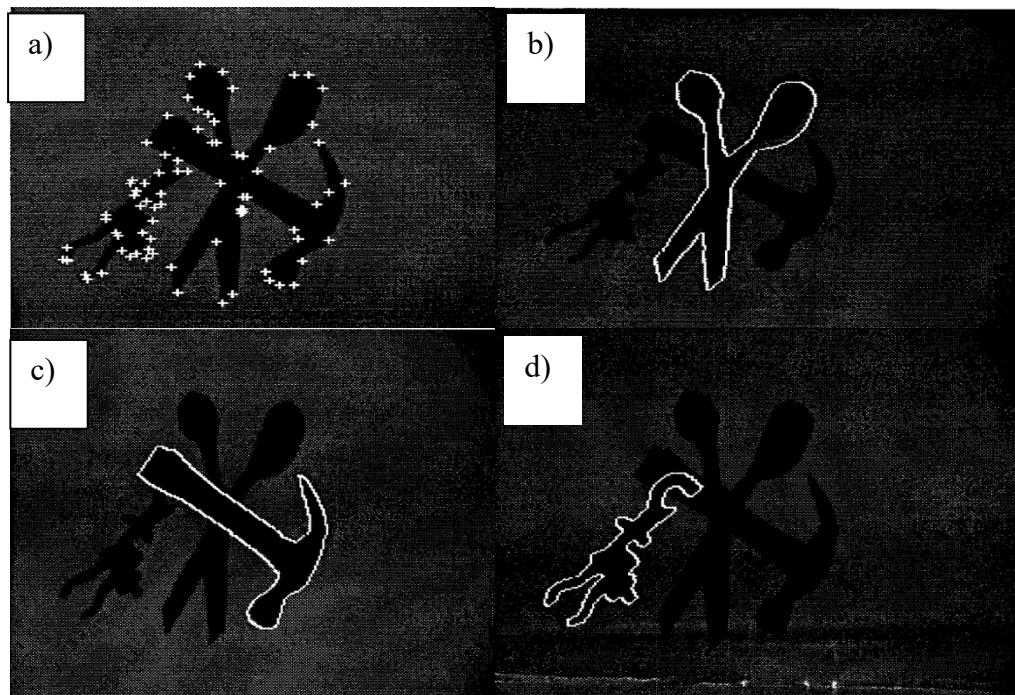
Figure 2.8: Ionic Polymer Metal Composite   (a) IPMC Prototype in actuation (Hunt et al., 2018). (b, c and d) Mechanical actuation of an IPMC mechanism (Madden et al., 2004).

2.2   Object Recognition Techniques

2.2.1   Hopfield Network
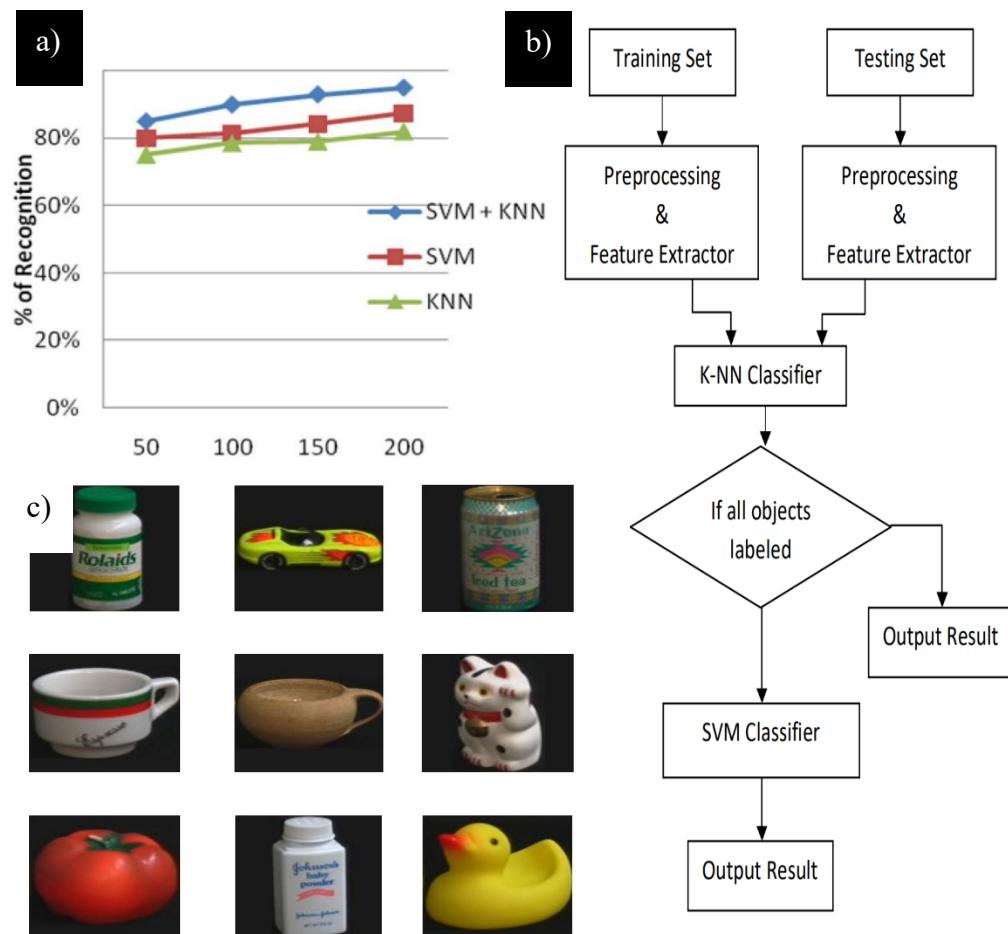
Hopfield Neural Networks can be used as a 2D object recognition technique due to their fast and more robust object recognition method by reducing computational costs(Young et al., 1994). This technique matches all the objects against all the object models within the model database simultaneously by taking advantage of the parallelism of the neural network where the objects can be touching or overlapping(Nasrabadi et al., 1991). A Hybrid Hopfield Network (HNN), which combines the advantages of a Continuous Hopfield Network (CHN) and a Discrete Hopfield Network (DHN) can also be applied. It provides high fault tolerance and better occlusion or edge detection(Kim et al., 1996). Occlusion occurs when two or more objects overlap or touch in an image, as shown in Figures 2.9. Hopfield Networks are also used as a high-resolution technique in acoustic imaging(Winters, 1988) and microwave imaging(Farhat, 1989).

Figure 2.9: Hopfield Network (a)Occluded test scenario with three overlapping objects, feature points are marked by '+' . (b)-(d) The coordinate transformation of each recognised models is evaluated and superimposed on the input image by applying a Hopfield Neural network (Kim et al., 1996).

2.2.2 Support Vector Machines (SVM)
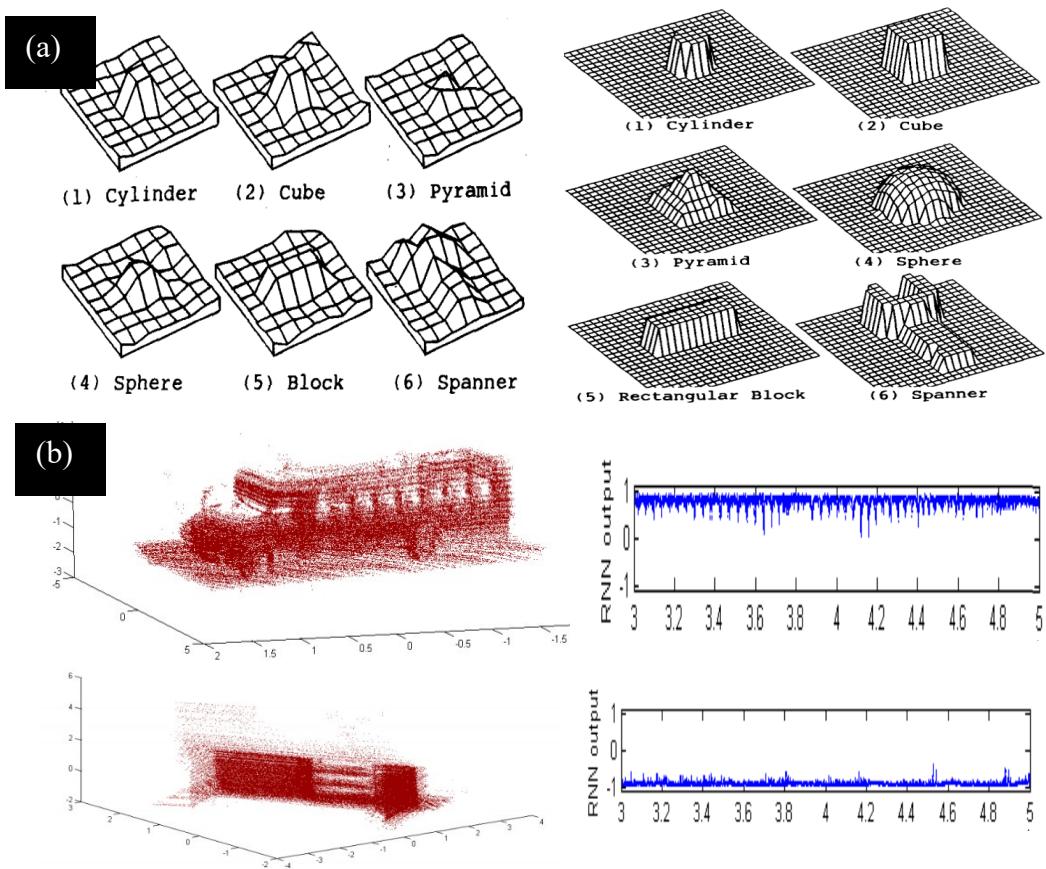
The Support-Vector network is a supervised learning network used for high generalisation group classification problems (Cortes et al., 1995). SVMs have been a standard tool to use when solving classification problems because they can learn any training set perfectly given an appropriate choice of kernel parameters and their loss function is convex making it easier to reach a global minimum (Huang et al., 2006). However, SVM training and testing is computationally intensive and tuning the kernel parameter is a complicated procedure (Zaidi et al., 2010). SVMs can also be combined with other neural network algorithms such as K Nearest Neighbour (KNN) to create a hybrid and more efficient algorithm (Muralidharan et al., 2011).

Figure 2.10:Support Vector Machine (a) Performance of each neural network algorithm and the hybrid of both algorithms (b) Model flowchart for an SVM + KNN object recognition (c) Example images of trained objects (Muralidharan et al., 2011).

2.2.3   Recurrent Neural Network

Recurrent Neural Networks (RNN) are widely used to model sequential data such as text and sound (Visin et al., 2015). This is because RNN can store a state (memory) of previous inputs and can process them sequentially. However, computer vision through light and sound data (LIDAR and ultrasonic) to recognise 3D objects have been achieved with an RNN (Prokhorov, 2009 & Watanabe et al., 1992). RNN can be trained by using gradient descent, which finds the minimum of a function. This can be used to minimise the error of the function. Gradient descent works by changing neuron's weights in proportion to the differential error with respect to each neuron's weight in each iteration.
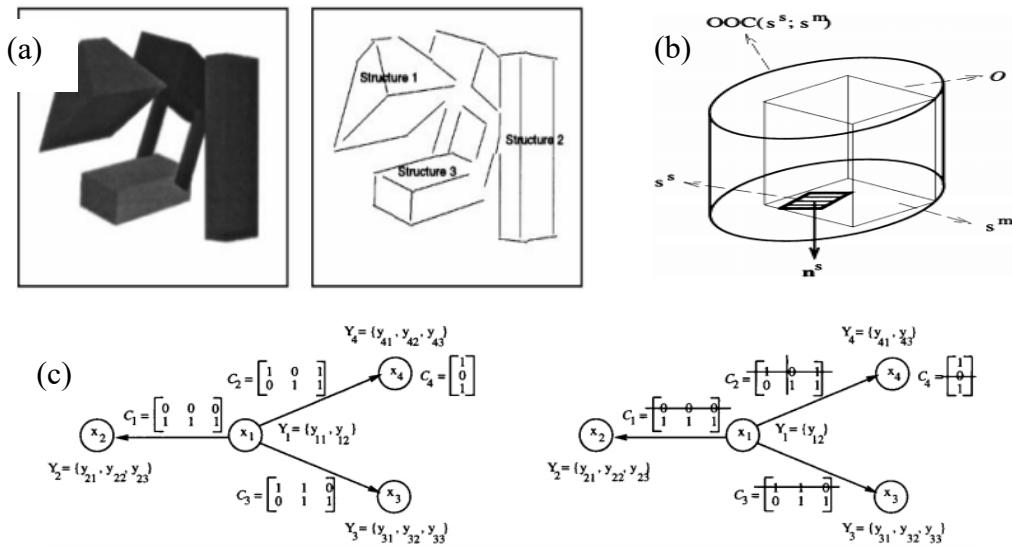
Figure 2.11:Recurrent Neural Network (a) 3 Dimensional images using acoustic imaging (left) and improved images using RNN processing (right) (Watanabe, 1992). (b) 3 Dimensional images obtained using LIDAR imaging of a vehicle and a non-vehicle structure (left) and the RNN prediction model of vehicle classification, where values above 0 indicate a vehicle and values below 0 indicates a non-vehicle (right) (Prokhorov, 2009).

2.2.4   Constrain-Satisfaction Approach

To recognise objects requires satisfying both object-rigidity and scene-data constraints to features extracted from a test model and a model object (Boshra et al., 1999). By integrating different types of scene features or data types of different dimensions (visual and tactile), a Constraint-Satisfaction Problem (CSP) can be formulated to unify the different data sets (Boshra, et al., n.d.). This problem can be solved by applying a data drive tree search by using local constraints (Boshra et al., 1999) as well as a local-consistency enforcing (LCE) technique which reduces the uncertainty between the scene and model features (Boshra, et al., n.d.). Constraints such as visual, tactile unary, and visual/tactile binary are applied to the scene constraint network to ensure that the visual/tactile feedback is bounded. Examples of visual constraints include (Boshra, et al., n.d.):
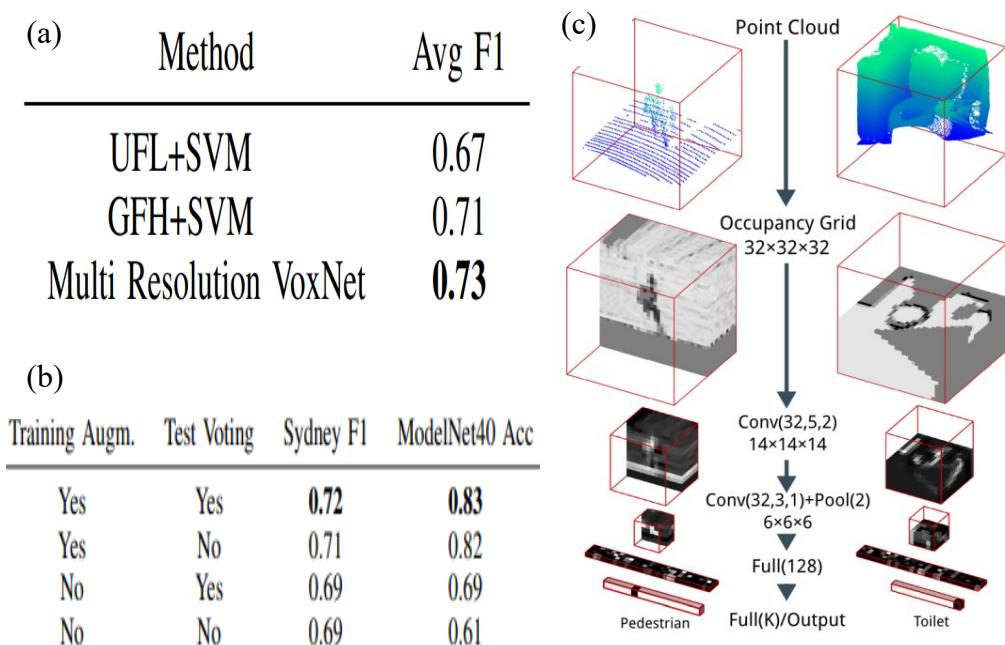
1. Same Edge Constraints: Points that corresponds to collinear edges.
2. Parallel- Edge constraint: Points corresponding to edges that are parallel.
3. Same-Junction Constraints: Visual edges that belong to the non-T junction.
4. Same-Object Constraints: Points that are perceived to belong to the same object.

Figure 2.12:Constrain-Satisfaction Approach (a) 3 synthetic objects and the extracted perceptual structures (b) Illustration of an object-occupancy constraint (c) An Illustration of LCE, where the left diagram shows a CSP and the right diagram shows a CSP after one iteration of LCE (Boshra et al., 1999).

2.2.5 Convolutional Neural Network

Convolutional Neural Networks (CNN) have been widely used to analyse visual imagery (Valueva et al., 2020). A convolutional neural network has a similar architecture as other neural networks in which they contain an input layer, an output layer, and sometimes multiple hidden layers. Convolution networks use convolution instead of matrix multiplication in at least one of their layers (Goodfellow et al., 2016). CNNs can realise the features and the extraction of visual signs in an object with minimal human interactions, which is why CNN is mainly used in image recognition and object detection (Valueva et al., 2020). CNN works by convoluting features within a set of data and applying a local or global pooling algorithm to improve the underlying computation efficiency. Range and LiDAR sensors have been used in combination with a CNN algorithm to act as a robust object recognition technique by obtaining a 3D point cloud data and applying an occupancy grid with a convolutional layer and a pooling algorithm to reduce the dimension of the data(Maturana et al., 2015).

(a)

Method	Avg F1
UFL+SVM	0.67
GFH+SVM	0.71
Multi Resolution VoxNet	**0.73**

(b)

Training Augm.	Test Voting	Sydney F1	ModelNet40 Acc
Yes	Yes	**0.72**	**0.83**
Yes	No	0.71	0.82
No	Yes	0.69	0.69
No	No	0.69	0.61

(c) Point Cloud

Occupancy Grid
32×32×32

Conv(32,5,2)
14×14×14

Conv(32,3,1)+Pool(2)
6×6×6

Full(128)

Pedestrian   Full(K)/Output   Toilet

**Figure 2.13:Convolutional Neural Network (a)Comparisons of different recognition methods (UFL+SVM (De Deuge et al., 2013) and GFH+SVM(Chen et al., 2014)) with a CNN network (Multi Resolution VoxNet) and the average test accuracy (Avg F1) (b) Effects of Rotation Augmentation and Voting on the test accuracy on different datasets. (c) A CNN based object recognition architecture (VoxNet) (Maturana et al., 2015).**
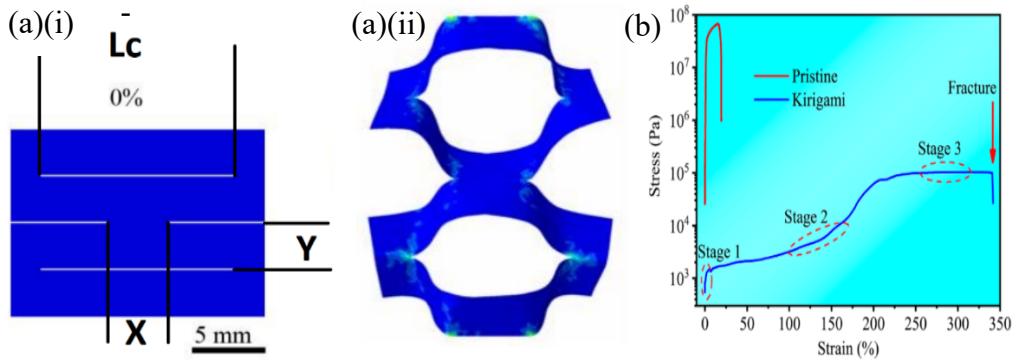
2.3   Kirigami structures

The art of kirigami consists of folding and cutting paper into highly intricate structures. Kirigami patterns have been discovered and implemented in several studies due to their special properties that enable great stretchability. For example, Hunt et al., 2018 and Rafsanjani et al., 2018 have integrated kirigami structures into actuators to allow for a higher strain limit and provide linear displacement for the applications of soft robotics. Not only that, Song et al., 2015 studied the application of stretchable lithium-ion batteries using the concept of kirigami; and Li et al., 2020 demonstrated highly stretchable solar cells, which is also fundamentally based on the kirigami concept. All these applications require carefully designed and optimised kirigami structures to ensure maximum stretchability with minimum wear. Hence, the optimum cutting patterns are key to determine the property of a kirigami structure.

Several cut designs have been studied. One of the more typical kirigami cuts is the linear cut. Figure 2.14(a) shows an example of a linear cut that consist of horizontal cuts along a strip of material and Figure 2.14(a)(ii) shows the elongated structure of the kirigami pattern. This pattern is characterised by the length of each cut, L_C, and the distance of the horizontal and vertical (x and y) directions of the cuts (Shyu et al., 2015). The cut patterns allow for a higher uniaxial strain than an uncut structure, as shown in Figure 2.14(b).

Figure 2.14(b) also shows the different stages of elongation of a kirigami model. The stages are described as (Li et al., 2020):

- Stage 1: Mechanical deformation of the kirigami structure is similar to that of the uncut film and undergoes elastic deformation along the loading plane.

- Stage 2: The kirigami structure converts from a stretching motion to a bending motion by rotation of the struts. The new film exceeds its critical buckling constant and fractures.

- Stage 3: When the loading increases, the stresses remain unchanged before fracturing as the struts begin to align in the load direction.
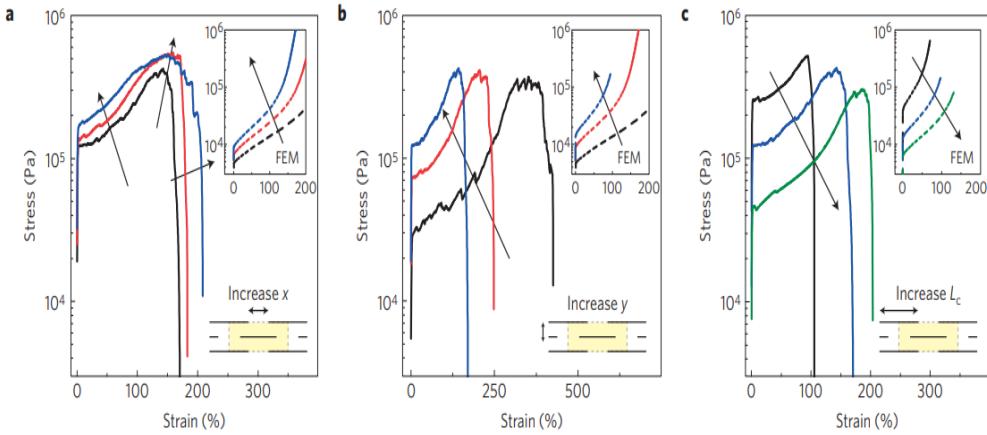
Figure 2.14: Kirigami Structures (a)(i) Illustration of a linear cut kirigami structure with the defining characteristics labelled LC, X and Y(Li et al., 2020). (a)(ii) Stress distribution of the elongated kirigami pattern (Elongation = 50%) (Li et al., 2020). (b) Stress-strain curves of an intact(pristine) and a linearly cut(Kirigami) cellophane substrate. The Kirigami structure undergoes three stages of elongation before fracturing(Li et al., 2020).

2.3.1 Optimisation of Kirigami structures.

2.3.1.1 Manipulation of characteristics
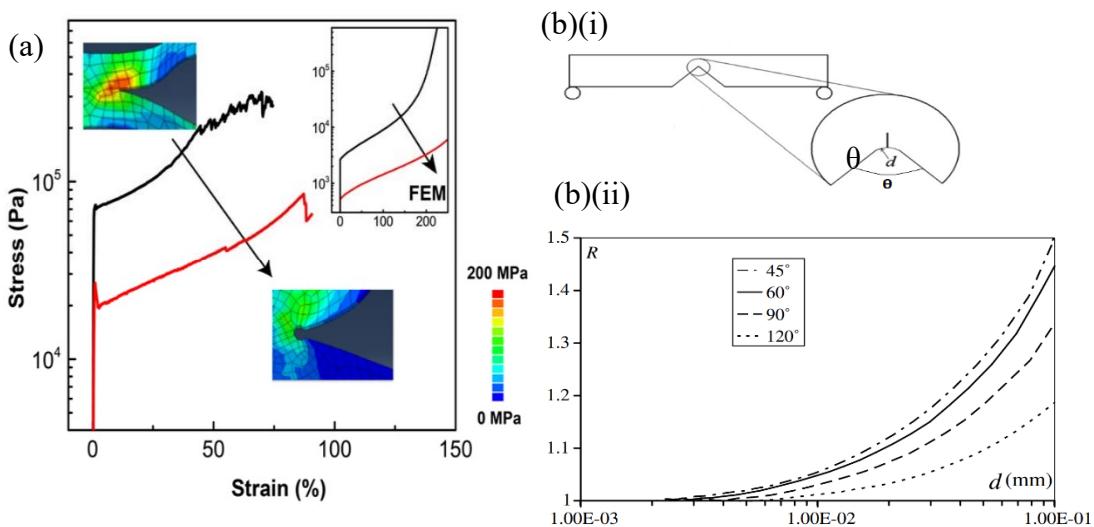
By manipulating the different characteristics (L_C, X and Y) of the typical kirigami structure, different stress-strain curves can be plotted, as shown in Figure 2.15. Therefore, to achieve a max strain with minimal stresses, the three variables may be tweaked to obtain an optimal stress-strain balance. The plots in Figure 2.15 was measured based on a typical straight cut kirigami structure.

Figure 2.15: Manipulation of Characteristics (a)-(c) Experimental Stress-strain curve and finite element modelling (FEM) results on the top right corner of each main figure. The figures show manipulation of X, Y, and LC, respectively. The purpose of the single sided arrow is to show the different outcomes when the respective characteristics' distance is increased (Shyu, 2015).

2.3.1.2 Sharp notch blunting
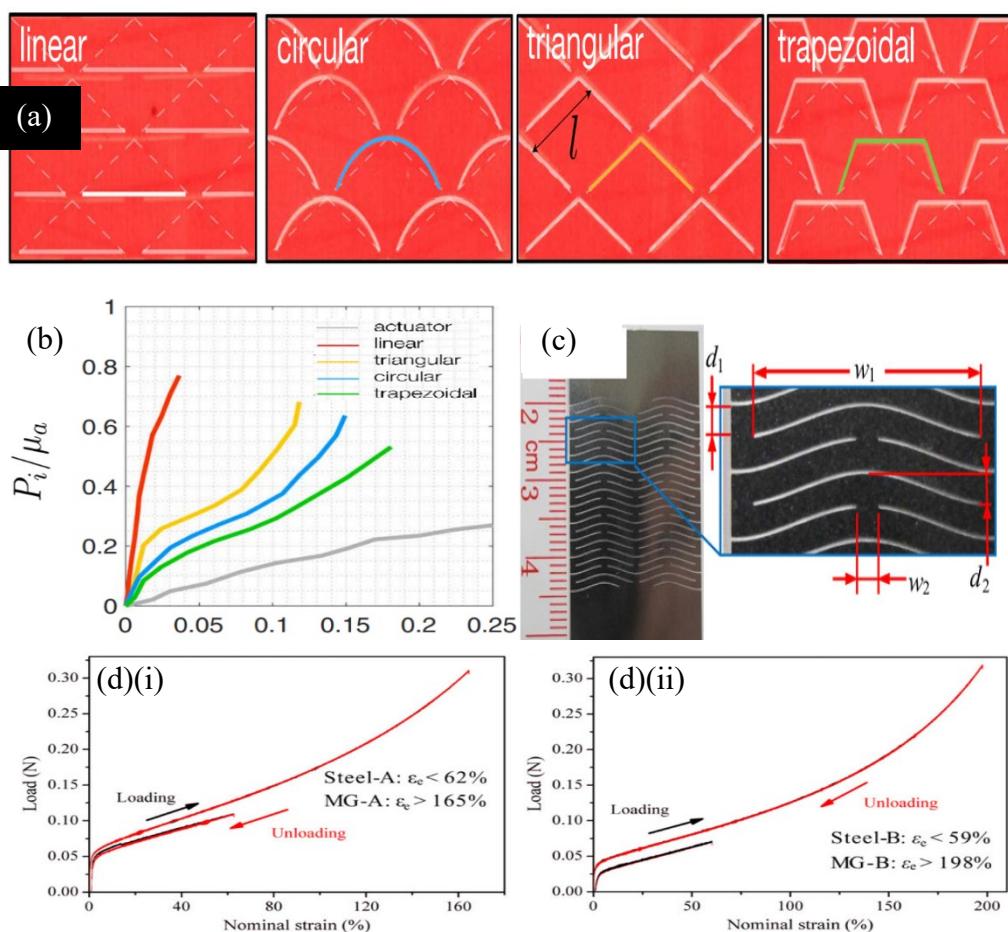
In the study of the propagation of cracks in materials, the fractures in 'V' notches can be hindered or prevented by blunting the notch to allow for a more distributed stress around the notch (El Haddad et al., 1979). Therefore, for a linearly cut kirigami structure, the vertices experience concentrated stress. The addition of a round hole at the vertices can further distribute the stresses more evenly, as shown in Figure 2.16(a). The toughness of the material is improved as the notch radius increases, as shown in Figure 2.16(b)(ii)(Leguillon et al., 2007).

Figure 2.16: Sharp Notch Blunting (a)Experimental stress-strain curve and finite-element modelling (FEM) results on the top right corner of the main figure (Shyu et al., 2015). The plot shows the sample with(red) and without(black) the blunting of the vertex. (b)(i) The formation of a short crack at the V notch's vertex (Leguillon et al., 2007). (b)(ii) A plot of apparent strengthening improvement, R with the notch root radius, d (in mm) for different angles of notches (refer to Figure 2.16(b)(i) for visualisation)(Leguillon et al., 2007).

2.3.1.3 Shapes of kirigami pattern
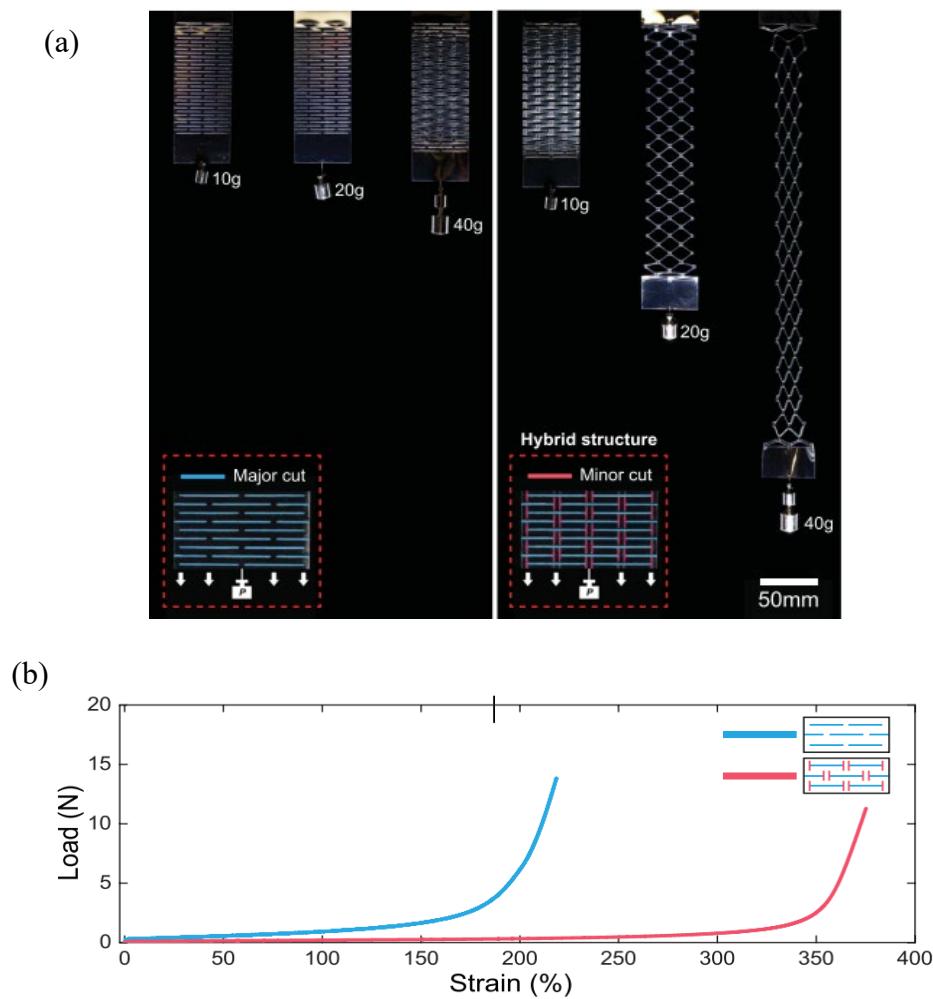
Several studies have proposed different kirigami structure design patterns to optimise the strain conditions and maximising elongation. For example, A paper published Rafsanjani et al., 2018 reported that a trapezoidal repeating pattern inspired by snake scales provides the best pressure-elongation curve (Figure 2.17(b)) among the different cut patterns as shown in Figure 2.17(a). Chen et al., 2018 reports that a wave-like cut (Figure 2.17(c)) allows for a higher strain before buckling (Figure 2.17(d)(ii)) as compared to a typical straight cut (Figure 2.17(d)(i)) in both stainless steel structure and the Metallic Glass(MG) structure.

Figure 2.17: Shape of kirigami pattern (a) Different patterns of cuts that allows for different characteristics (Rafsanjani et al., 2018). (b) The pressure-elongation plot of the different types of cuts in which the pressure is normalised by the shear modulus of the actuator(Rafsanjani et al., 2018). (c) Wave-like cut design and its respective characteristics (d1, w1, w2, d2)(Chen et al., 2018). (d) Load-strain curve of a typical straight cut configuration(i) and the curve of a wave-like cut(ii) (Chen et al., 2018).

2.3.1.4 Hybrid structure

A combination of cuts can be made in order to reduce the stiffness further and increase the maximum buckling stress of a material. The addition of minor cuts to typical linearly cut kirigami structure as shown in Figure 2.18(a) can reduce the stiffness by a factor of 30 as well as the increasing the maximum buckling stress by a factor of 2 relative to single-incision patterns as shown in Figure 2.18(b) (Hwang and Bartlett, 2018).
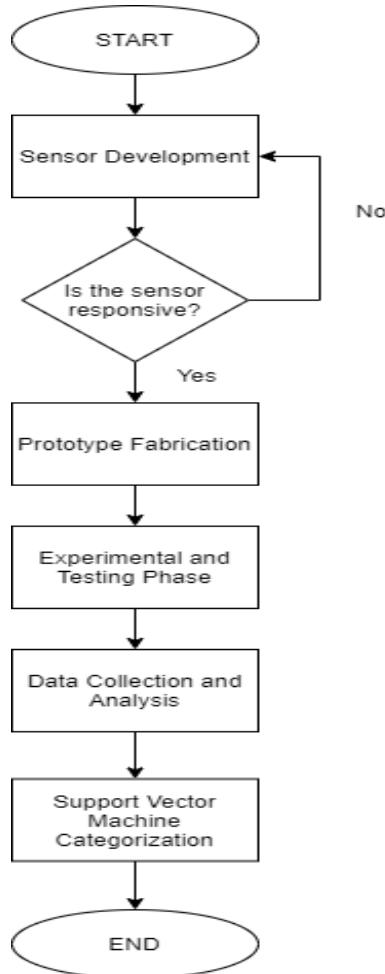
Figure 2.18: Hybrid Structure (a) kirigami sheets with only major cuts (left) and hybrid structure with major and minor cuts (right) (Hwang and Bartlett, 2018).(b) Load-strain plot for major cuts(blue) and major and minor cuts(red) (Hwang and Bartlett, 2018).

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1   Overview

The overview of the entire project is visualised in the flow chart as shown in Figure 3.1.

Figure 3.1: Illustration of the project flow

Firstly, the development and experimentation of different types of sensors were done. Once a responsive sensor was found, the prototype was fabricated using the sensor mentioned above. The characterisation and sensitivity testing of the sensor was also done. Finally, the sensor's resultant data was then collected and labelled to be used to train a support-vector network to recognise different object shapes.
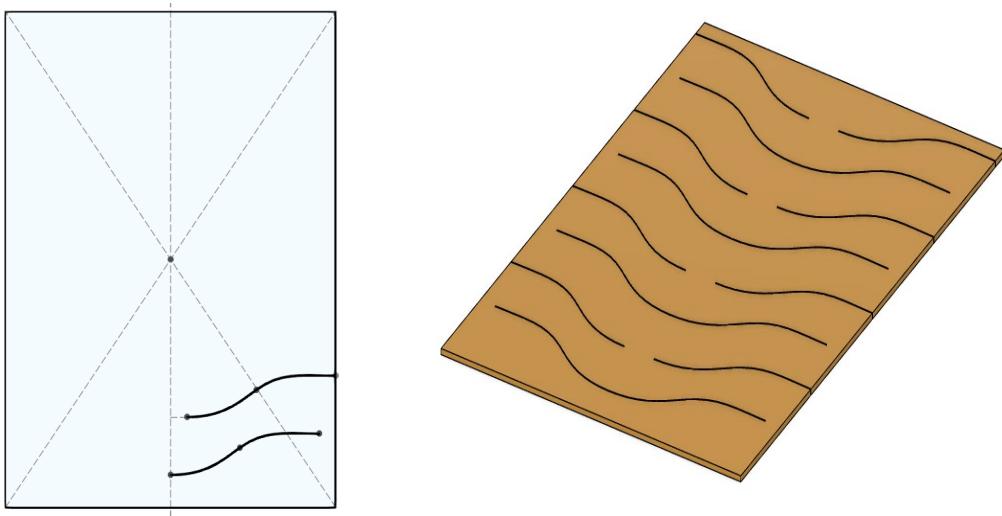
3.2   Sensor Development

Initially, an IPMC based active bend sensor with integrated kirigami cutting was planned to be studied and used in this project. Hence, the kirigami structure's dimensions were designed, and the optimisation of the kirigami structure was done with Finite Element Analysis (FEA) to observe the stress-strain characteristics of the structure. However, due to the imposed movement control order caused by the Covid-19 pandemic during the experimental stage of this study, further fabrication of the prototype using IPMC compound could not be completed in the lab. As a result, this study only provides the FEM simulation of the kirigami structures without hardware verifications.

Subsequently, a resistive based flex sensor was used instead as it was readily available. The resistive sensor was characterised and its sensitivity was investigated.

3.2.1   Structural Design of Kirigami Structure
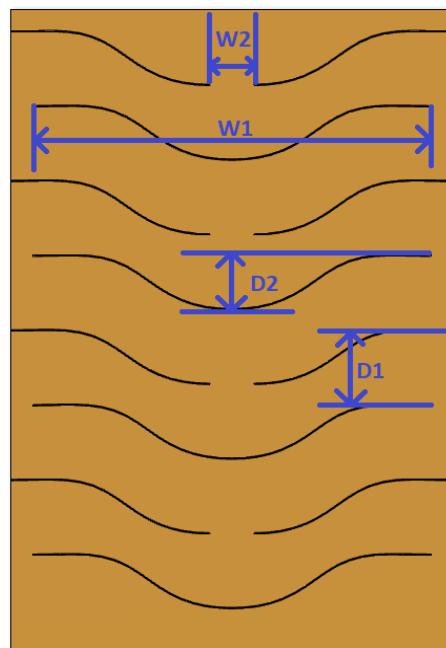
A kirigami structure was first designed in Autodesk Fusion 360, a 3D CAD software capable of designing complicated 3D models. The design phase went through several iterations to make certain that the spacing between each structure's cuts is spaced evenly so that a consistent result can be obtained from the Finite Element Analysis in a later stage. In the end, the design is shown in Figure 3.2(b) is chosen due to its consistent result when simulating. A wave-cut was chosen based on its higher strain capabilities (Chen et al., 2018). Each of the wave structure is fully constrained to the model and modelled to be 0.05 mm thick. Figure 3.2(a) shows the initial sketch of the cuts before being mirrored and 'rectangular pattern' to form a repeating pattern along the length of the structure.

Figure 3.2: Structural Design of Kirigami (a)Initial sketch of kirigami structure. (b) 3D view of structure designed in Autodesk Fusion 360.

3.2.1.1   Structure Parameters

W1 is defined as the length of the center cut for each section; parameter W2 is the length of the space between two adjacent cuts; parameter D1 is defined as the length between the center cut, and the two adjacent cuts and parameter D2 is the height of the 'wave' of the cut as shown in Figure 3.3. To fully optimize the kirigami design characteristics, each of these parameters is manipulated and simulated to obtain the minimum possible theoretical stresses.

Figure 3.3: Illustration of the parameters in the kirigami structure.

This structure was designed to be 30 mm x 20 mm x 0.2 mm. The parameters for each of the structure in Figure 3.3 were varied three times to find the parameter's relationship to the stresses on the structure. Extra care was taken when modifying the parameters that the overall structure and number of cuts were not affected to prevent inaccurate results. The variations of lengths that were varied are as shown in Table 3.1.

Table 3.1: Variation of the parameters in the kirigami structure.

Parameters			
W1	**W2**	**D1**	**D2**
17 mm	2 mm	3.5 mm	3 mm
18 mm	3 mm	4.5 mm	4 mm
19 mm	4 mm	5.5 mm	5 mm

3.2.2   Finite Element Analysis
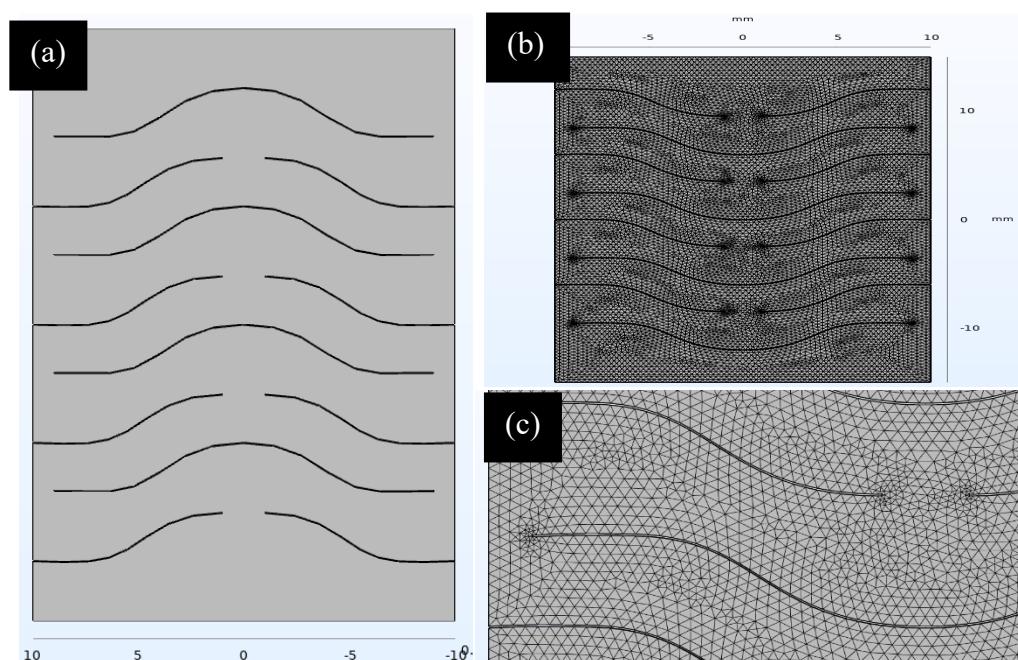
3.2.2.1   Introduction

Finite Element Analysis (FEA) is a method for solving boundary value problems using a numerical method. Typical studies that may fall under the boundary value problem may include structural analysis, mass transport, fluid flow, and heat transfer (Logan, 2011). A boundary value problem is a mathematical problem in which a set of dependant variables must fulfil a differential equation everywhere over a known range of independent variables (Hutton, 2004).

The analysis can be done by considering the different independent variables such as the material properties and geometric properties. The FEA is created by breaking down the structure into millions of polygons stitched together, known as a mesh. These smaller elements are then calculated individually based on the boundary conditions and combined to produce the whole structure's final result. Therefore, this can be used to produce a close approximation of loading stresses in our application.

28

3.2.2.2 COMSOL Multiphysics

In this study, FEA is done through COMSOL Mutiphysics 5.5, a mathematical modelling software designed to solving scientific and engineering problems. A 3D model was first designed through Autodesk Fusion 360, and exported into a stereolithography(stl) file format. Then, the file is then imported into COMSOL Multiphysics(Figure 3.4(a)), and a mesh is created using the 'finer' settings to allow for a more accurate analysis shown in Figure 3.4(b). A close-up image is shown in Figure 3.4(c) to capture the small polygons that make up the mesh. The deformation of the structure is then computed using the solid mechanics module in COMSOL.

Figure 3.4: COMSOL Multiphysics (a) Imported stereolithography(stl) file into COMSOL.(b) Mesh created using the 'finer' settings. (c) Close-up of the mesh.

3.2.2.3 Material and Configuration Properties

The material used in the COMSOL simulation has a density of 2.2 x 10^3 kg/m^3 with a yield strength of 2.18 x 10^7 Pa. Not only that, it has a thermal expansion coefficient of 1.7 x 10^{-4} strain/°c and a Poisson's ratio of 0.2. A boundary load pressure is exerted on one end of the kirigami structure while the other end is held constant. This will allow the kirigami structure to elongate perpendicularly to the cuts on the structure.
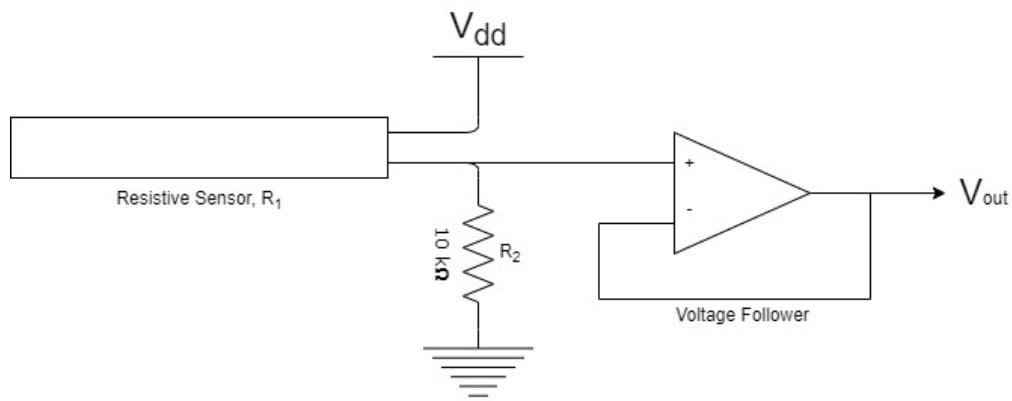
3.2.3   Bend Angle Characterization

3.2.3.1 Experimental Setup

To test the resistive strain sensor's bend angle response, a simple voltage divider circuit is assembled, as shown in the electrical schematic of Figure 3.5. A voltage buffer is added before the output stage to reduce the flex sensor's source impedance error. The resistance change within the flex sensor will cause a change in output voltage, V_{out} according to the voltage divider rule, where:

$$V_{out} = \frac{R_1}{R_1 + R_2}(V_{dd})$$

R$_2$ is fixed to be 10kΩ in our experimental setup to provide a readable range of output voltage to the microcontroller. The output voltage, V_{out} will be connected to an Arduino Uno to read and visualise the data. A C++ program(Appendix A) was written to read the analog voltage value from V_{out} and provide a graph to visualise the voltage change.
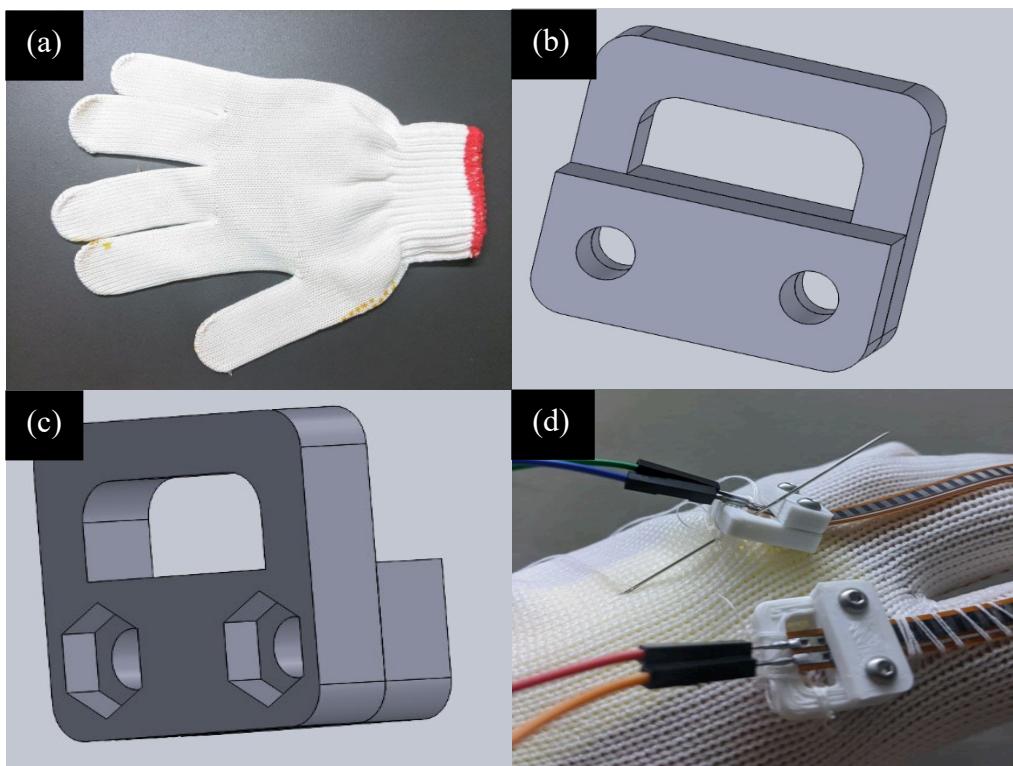
Figure 3.5: Experimental setup of a resistive strain sensor

The characterisation is done by bending the index finger to which the resistive sensor is attached and measuring the output voltage from the voltage divider. This allows a characterisation of the resistive range of the sensor and in turn allows for an estimated output voltage range.

3.3   Prototype Glove
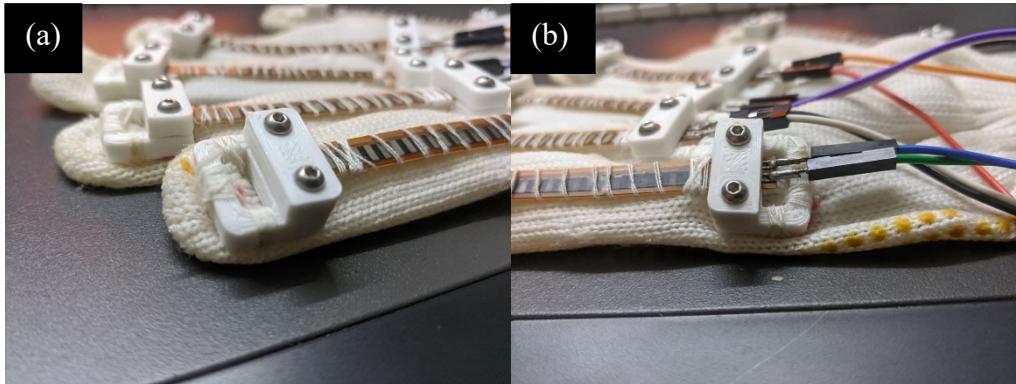
Once the flex sensor was fully characterised, a prototype glove was made from a total of 5 flex sensors, one for each finger. The flex sensors are attached to each finger's top and held on by a custom 3D-printed strain relief, allowing it to bend and stretch freely.

The glove chosen is a cotton gardening glove shown in Figure 3.6(a) which can be sewn to fit the flex sensor. The sensors are first attached on a 3D printed strain relief designed in Solidworks shown in Figures 3.6(b) and (c). The 3D printed strain relief is then sewn on to the glove, as shown in Figure 3.6(d).
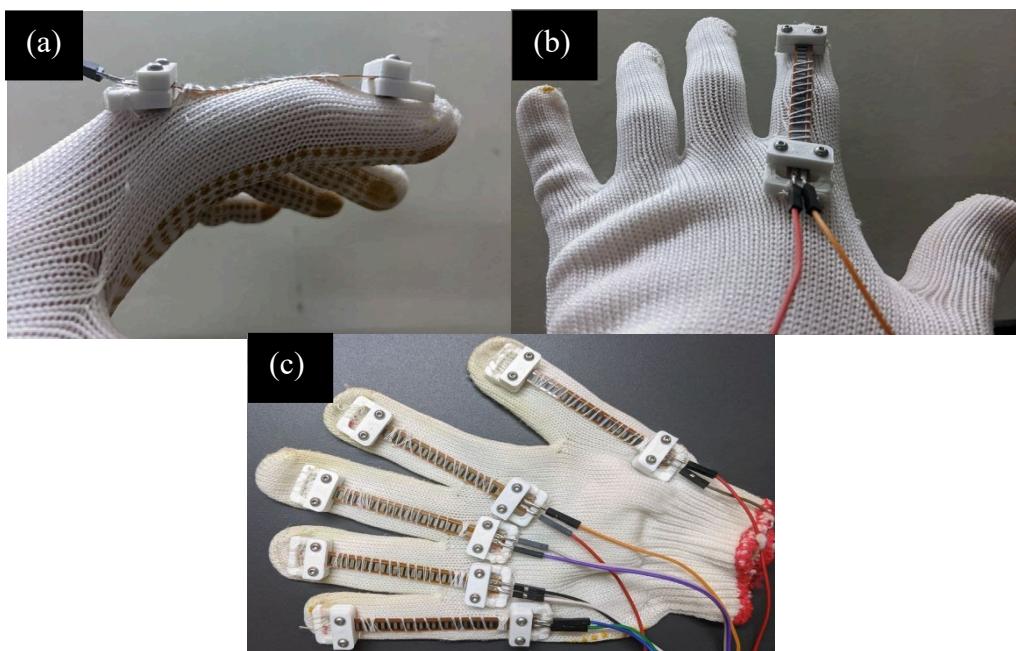
Figure 3.6: Attachment Methods (a) Cotton glove used as the prototype glove.(b)-(c) 3D CAD Drawing of strain relief designed in Solidworks. (c) Strain relief being sewn onto the glove to allow the sensor to be attached.

Each of the fingers consists of two strain relief clamps – one on the tip and another on the finger's base as shown in Figure 3.7(a) and (b), respectively. These clamps serve two purposes: fixing the flex sensor placement on the glove and providing strain relief when the sensor is bent.

Figure 3.7: Strain Relieve Clamps (a)-(b) Tip and base of the glove with sewn-on strain relieve clamps.

Once the clamps are sewn on, the flex sensors are attached to the clamps with two screws and the sensors are then sewn on to the glove, ensuring the sensor is as close to the finger as possible length wise as shown in Figure 3.8(a) and (b). The procedure is repeated four more times to attach all five sensors to each finger to create the glove prototype shown in Figure 3.8(c).

Figure 3.8: Prototype Glove (a)-(b)Fully sewn on sensor from side and front view respectively. (c)Finished prototype

3.4   Training Data Collection
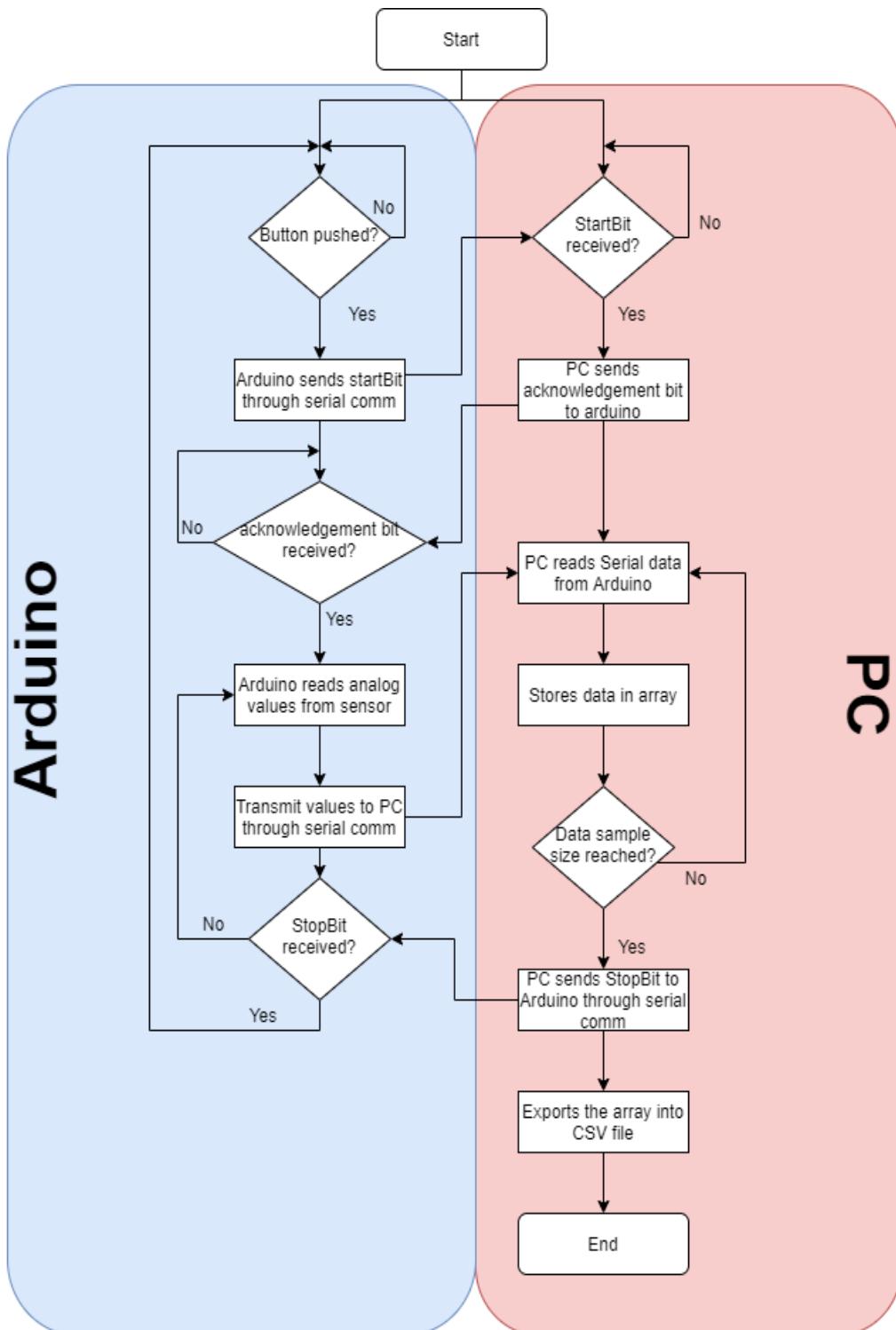
Figure 3.9:Programming Flowchart for Training Data Collection

Figure 3.9 shows the programming flowchart of the data collection procedure used to train the SVM model in this application. The program was written in Python language for the PC and C++ for the Arduino (the code is attached in Appendix B1 and B2 respectively).

The training was done on three objects with different shapes: a sphere, a cuboid, and a cylinder, as shown in Figure 3.10. The objects were designed in Solidworks and 3D-printed to ensure dimensional accuracy (Table 3.2) and provide a controlled way of collecting training data. The training data was obtained by wearing the glove, pushing the push button to enable data logging, picking an object up and releasing it to form a dynamically changing time-series graph. Each sample consists of 150 data points, and each object was sampled 200 times for a total of 30,000 data points for each object. The samples are then labelled to the shape of the object it belongs to and exported to a CSV file format with a distinct name to allow for easy debugging and file navigation in the training stages.

Figure 3.10: Objects used for training

Table 3.2: Object Dimensions

Sphere	Cuboid	Cylinder
Radius: 20mm	Length: 55mm Width: 55mm Height: 10mm	Radius: 23.5mm Height: 65mm

3.5   SVM Model Training
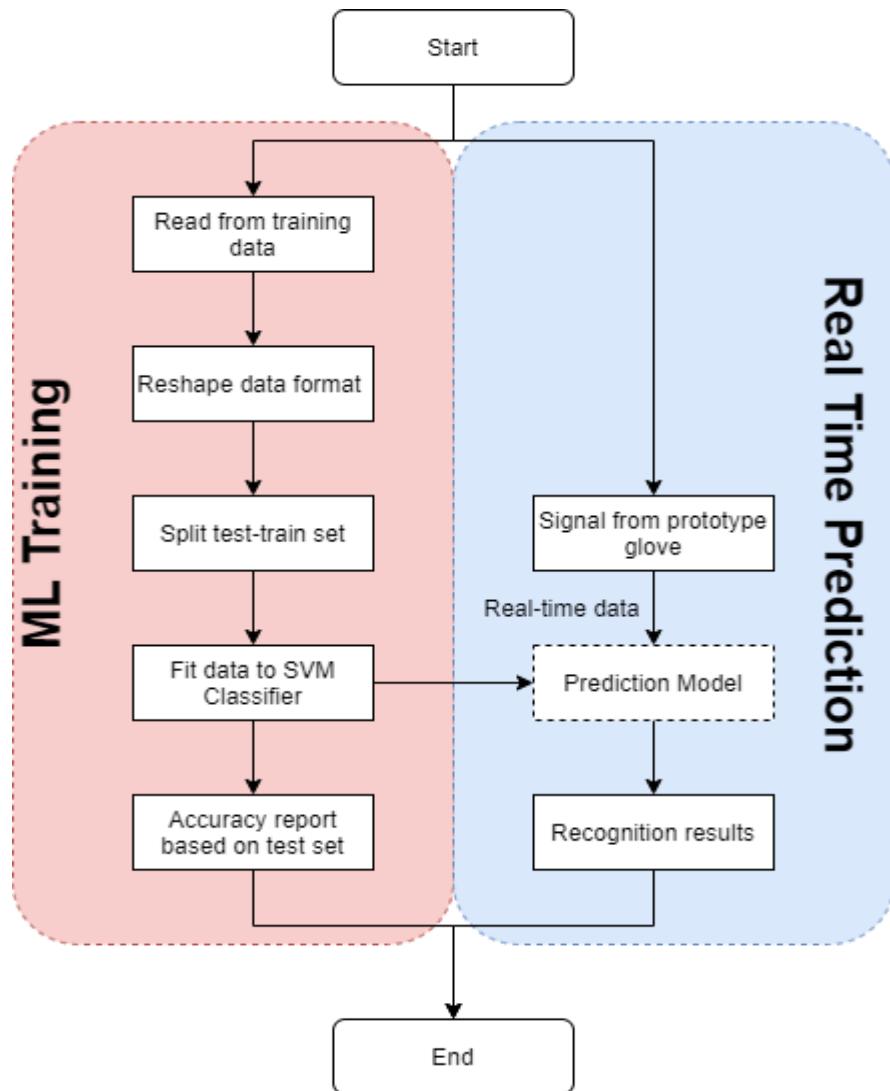
Figure 3.11: Programming flowchart for SVM model fit

After obtaining the training data for the three objects, an SVM classifier is implemented using Python's scikit-learn library. The data is split into two sets: a test set and a train set. The data in the train set is used to fit the SVM model and hence will contain a majority of the data sample; while the data in the test set is used to provide an unbiased evaluation of the performance of the model. In this study, the train-test set was split by 80% and 20%, respectively. The data was also stratified. The test-train sets consisted of evenly distributed labelled data of each object to prevent a biased prediction. The fitted SVM classifier can then be used to predict an object's shape on a real-time basis. The flow is as shown in Figure 3.11.

CHAPTER 4
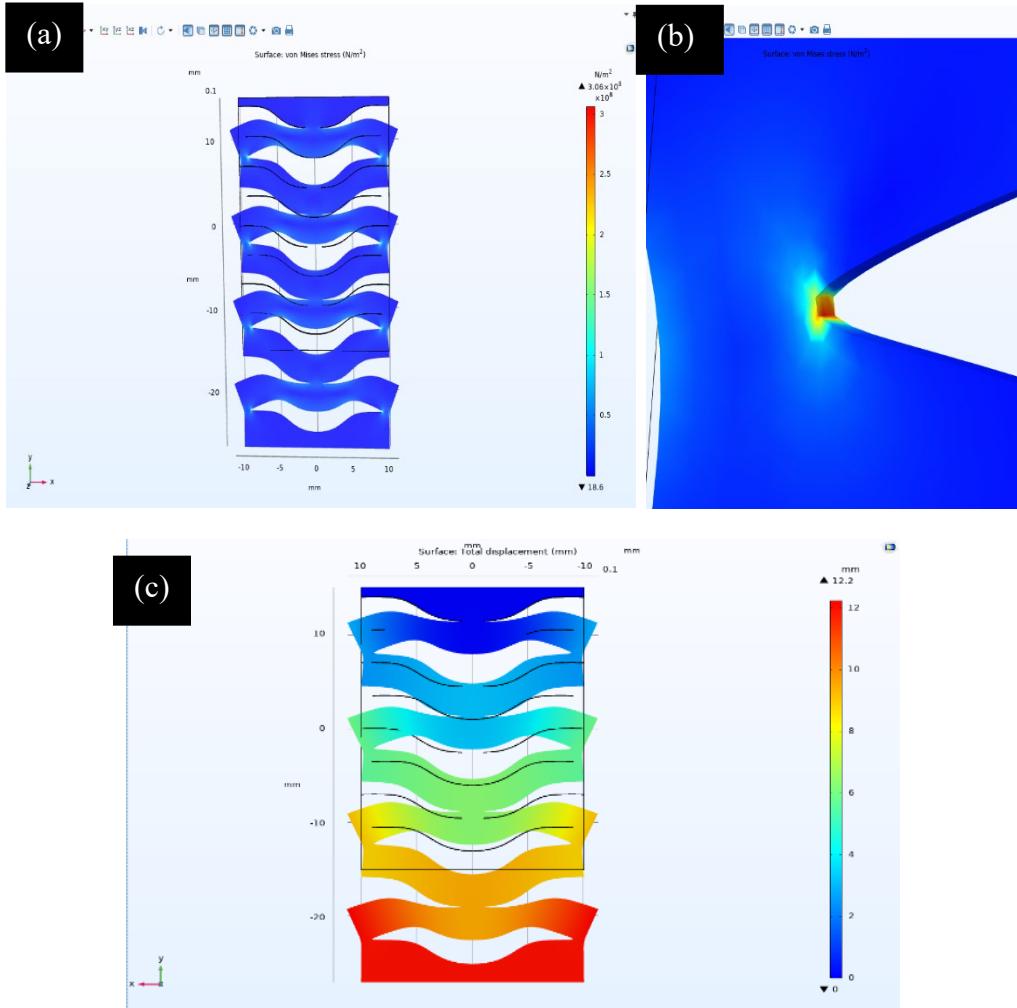
RESULTS

4.1   Kirigami

4.1.1   FEA Simulation of Kirigami Structure

The kirigami structure was analysed for its von Mises stress distribution along with the structure by varying the prescribed displacement. The von Mises stress analysis acts as a quantitative result to show the effective stress distribution of the structure when it is stretched, as shown in Figure 4.1(a). It is worth noting that according to the FEA, most of the stresses are concentrated on the vertices of each cut, as shown in Figure 4.1(b). The FEA results also show that the applied stresses are distributed uniformly throughout the structure and not concentrated on a single point.
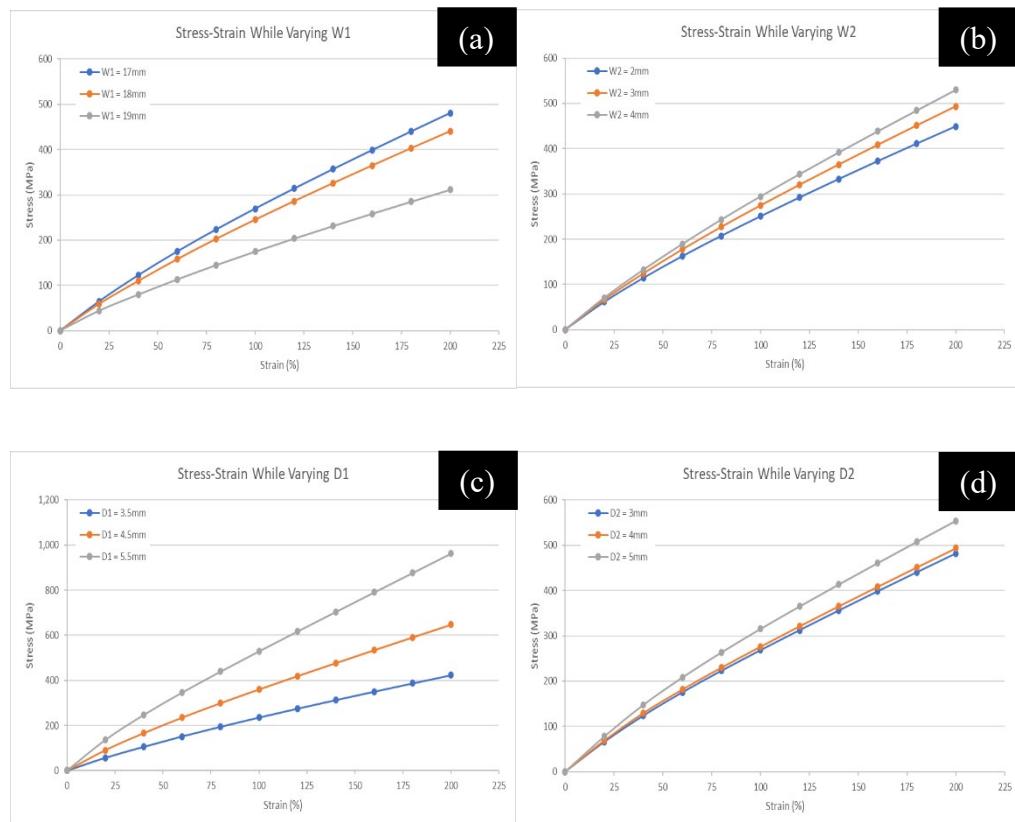
The prescribed displacement constraint allows for a constant displacement throughout the simulations. The responding variable is the stress exhibited by the different geometries. As the top of the structure has a fixed constraint added, the displacement of the structure is pulled down in a y-direction instead, as shown in Figure 4.1(c).

Figure 4.1: FEA Simulation (a)FEA of the kirigami structure. (b)Close-up of the notch of the structure to show stress distribution. (c)Prescribed displacement of 12mm.

4.1.2   Stress-strain plot

To investigate an optimal kirigami structure's design, a stress-strain relationship can be obtained from the structure. This can be obtained by systematically varying the kirigami deformation control parameters, namely W1, W2, D1 and D2 as defined in Figure 3.3 in the previous chapter. The stress-strain relationship can then be plotted for each of the parameters, as shown in Figure 4.2.

Figure 4.2: Stress-strain Relationship (a)-(d) Stress-strain relationships of a curved cut Kirigami structure with varying parameters.
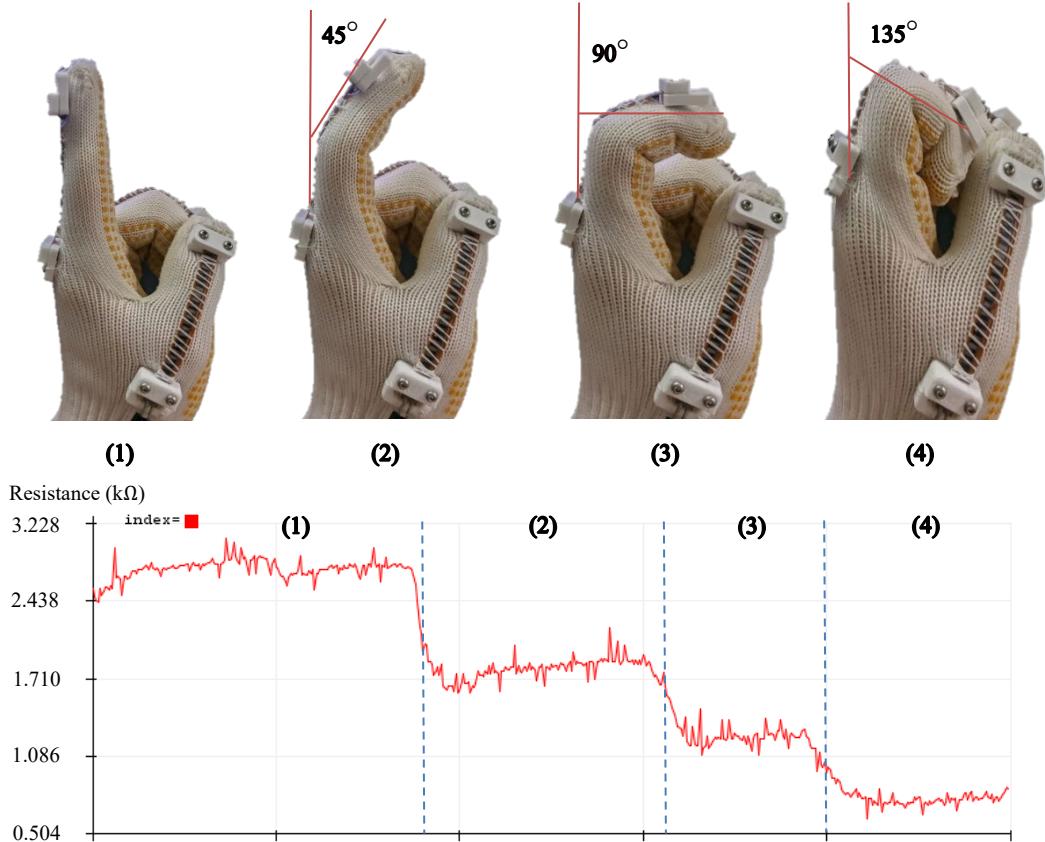
Predictably, the buckling load and the maximum extension of the structure are highly affected by the geometrical properties of the structure. The simulated results show that the increase of parameters W2, D1 and D2 leads to an upward shift in the graph, contributing to a higher buckling load but with a decrease in maximum extension.

On the other hand, an increase of parameter W1 responds to a downward shift in the strain-stress relationship. This means that as the W1 increases, the cuts essentially soften the material, leading to a higher extension of the structure. Thus, in general, it can be concluded that as the spacing between each cut increases, the structure will have higher rigidity and a higher critical buckling load. In contrast, increasing the cut length will make the material more flexible and increase its extensibility.

The FEA analysis of these models does not fully describe the material's entire property, such as the tearing and breaking of these models. Hence, the results obtained from the simulation merely acts as a validation of the experimental results.
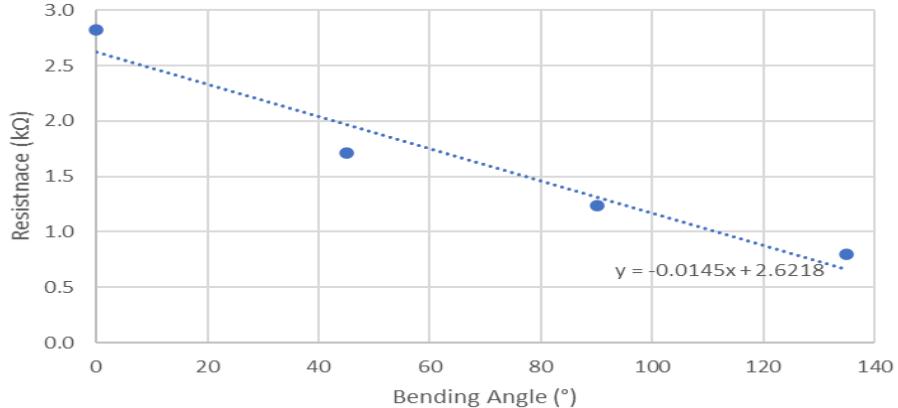
4.2 Resistive Strain Sensor Characterisation

The characterisation of the sensor is done by bending the index finger in a series of angles to produce a resultant output voltage on the serial plotter. Figure 4.3 shows the resultant voltage from four different bending angles of the index finger.

Figure 4.3: Resistive Strain Sensor Characterization. Demonstration of finger bending motion and the resultant resistance change

Motion (1) shows the finger in a straight manner, providing no bending motion on the sensor. With no bending motion, the sensor has a resistance of around 2.82 kΩ. Motion (2) is the index finger bending at a 45-degree angle with a corresponding resistance of around 1.71 kΩ. Similarly, Motion (3) and (4) has an average resistance of around 1.236 kΩ and 0.799 kΩ, corresponding to a bending angle of 90 and 135 degrees, respectively.

Figure 4.4: Relationship of resistance change with the bending angle.

From Figure 4.4, it is clear that the full range of the index finger movement corresponds to a change in resistance that ranges linearly from 0.799 kΩ to 2.82 kΩ with a resistive sensitivity of 0.0145 kΩ/° .

4.3   SVM Training

The data collection for training the SVM model consists of collecting the output voltage from five fingers and compiling it into a CSV format. The cuboid and the cylinder consist of multiple ways to be held, so the training data was separated into the number of ways each object is held. The total number of samples of each object is then divided equally into each grasping position. For example, for the cuboid, which consists of two ways it can be held, each grasping position is recorded 100 times for a total of 200 training samples of that object.

Figure 4.5 shows one of the samples of each object and the number of ways the object can be grasped. The y-axis on each output graph is a 10-bit analog value ranging from 0-1023. It can be mapped into a corresponding voltage by:

$$V_{output} = \frac{X_{analog}}{1023} \quad (5)$$
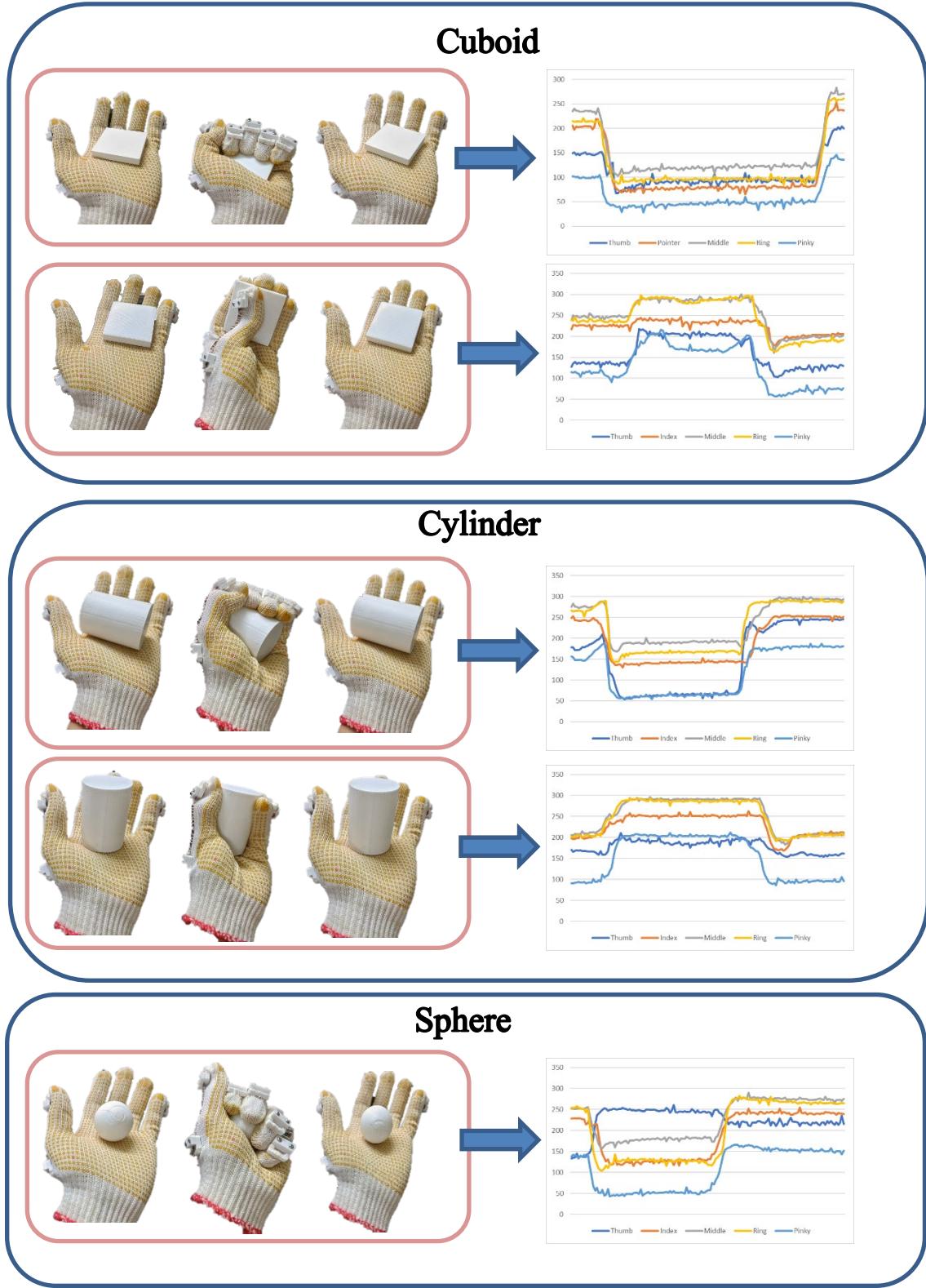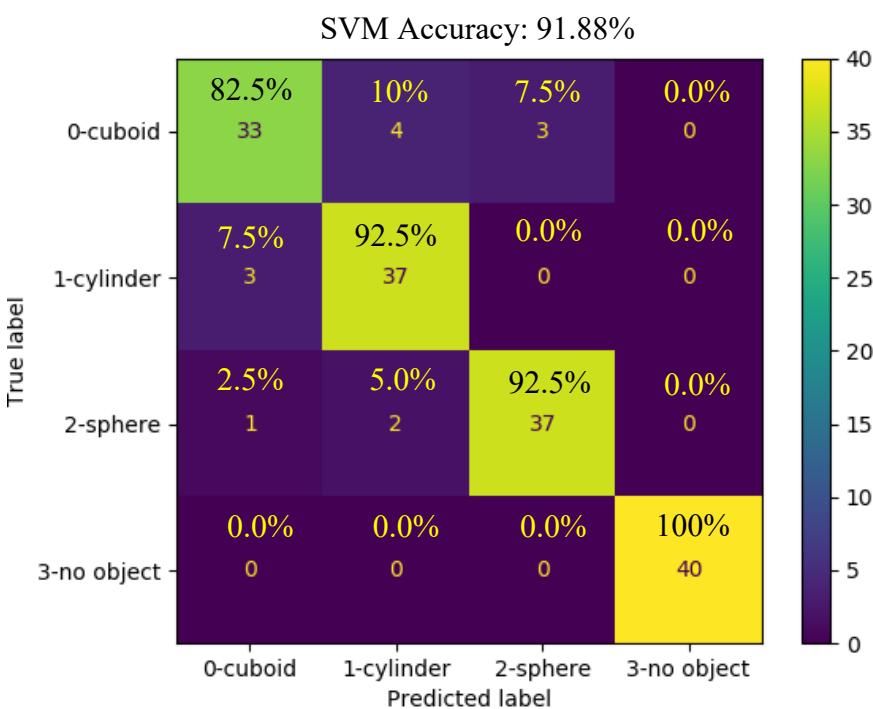
Figure 4.5:Sample output data of each shape.

4.4   SVM Fitting
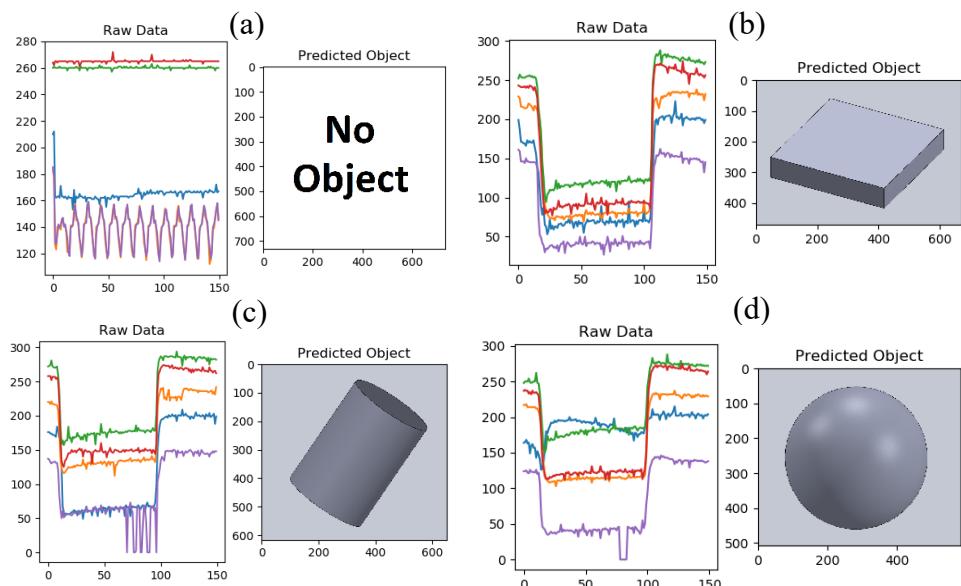
The SVM classifier used in this model is a C-support Vector Classification which supports multiclass classification. The confusion matrix of the SVM models in Figure 4.6 shows that this model can assist the glove in achieving more than 91% accuracy in recognising the shape of objects. The kernel parameter, C, which is used to determine the possibilities of misclassification, was remained at the default and not optimised. Each object has 160 training samples and a testing sample of 40 samples (80% to 20% split). The baseline (not holding any objects) has a 100% prediction rate and the highest error appears to be the cuboid with 82.5% true predictions with 7 samples being predicted wrongly. This can be attributed to the lack of training samples and the accuracy might increase with higher training samples.

Figure 4.6: Confusion Matrix. Confusion matrix derived from the SVM model with 40 tests for each object, including the baseline

4.5 Real-time prediction

The real-time prediction of the glove can be made by incorporating real-time input from the sensor into the trained SVM model. A button is pushed, sending a signal to the Arduino to start reading the sensor output and transmitting it to the Python program via serial communication. The SVM then takes the input data, parses it, and makes a prediction based on the trained samples. There are four possible outputs depending on the inputs, as shown in Figure 4.7. The Python program can be found in Appendix C.

Figure 4.7: Real-time Prediction (a)-(d) Raw real-time data provided by the glove with the correct corresponding predicted object shape.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, this study presented a prototype of an object recognition glove based on soft sensors. It was found that the resistance of the resistive flex sensor has an effective range of 0.799 kΩ to 2.82 kΩ with a sensitivity of 0.0145 kΩ/° when attached on a finger and bound to finger movements. Moreover, a prototype glove was fabricated using these flex resistive sensors, and an SVM model was applied to recognise the shape of an object. The SVM model successfully predicted the test set with a 91.88% accuracy. Hence, the objectives and the aim outlined in this project were achieved.

5.2 Future Works and Recommendations

To enhance the performance of the machine learning model, hyperparameter tuning can be applied to the SVM to allow for a higher accuracy prediction. Not only that, more training data can be provided to the model to enhance the categorisation capabilities of the model. This is evident in a study done by Zhu et al. in which they were able to obtain up to 96% accuracy with shape recognition by providing more training samples and tuning the penalty parameter(Zhu et al., 2020).

Moreover, a self-powered IPMC sensor could be used to substitute the resistive flex sensor used in this project. Due to the Covid-19 pandemic and the mandatory lockdown imposed within the country, the use of IPMC in this study was halted as it wasn't freely available and had to be fabricated in the lab. The change to an IPMC-based sensor with introduced kirigami cutting would bring several advantages, including a self-powered nature of the sensor and the increased stretching ability of the sensor.

REFERENCES

Boshra, M., & Hong Zhang., n.d. A constraint-satisfaction approach for 3D vision/touch-based object recognition. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots. doi:10.1109/iros.1995.526243

Boshra, M., & Zhang, H., 1999. A Constraint-Satisfaction Approach for 3-D Object Recognition by Integrating 2-D and 3-D Data. Computer Vision and Image Understanding, 73(2), 200–214. doi:10.1006/cviu.1998.0730

Cai, L., Song, L., Luan, P., Zhang, Q., Zhang, N., Gao, Q., … Xie, S., 2013. *Super-stretchable, Transparent Carbon Nanotube-Based Capacitive Strain Sensors for Human Motion Detection*. Scientific Reports, 3(1). do:10.1038/srep03048

Calvert, P., Duggal, D., Patra, P., Agrawal, A., & Sawhney, A., 2008. Conducting Polymer and Conducting Composite Strain Sensors on Textiles. Molecular Crystals and Liquid Crystals, 484(1), 291/[657]–302/[668]. doi:10.1080/15421400801904690

Chen, S. H., Chan, K. C., Yue, T. M., & Wu, F. F., 2018. *Highly stretchable kirigami metallic glass structures with ultra-small strain energy loss. Scripta Materialia, 142, 83–87.* doi:10.1016/j.scriptamat.2017.08.037

Chen, S., Cao, Y., Sarparast, M., Yuan, H., Dong, L., Tan, X., & Cao, C., 2019. Soft Crawling Robots: Design, Actuation, and Locomotion. Advanced Materials Technologies, 1900837. doi:10.1002/admt.201900837

Chen, T., Dai, B., Liu, D. and Song, J., 2014, June. Performance of global descriptors for velodyne-based urban object recognition. In 2014 IEEE Intelligent Vehicles Symposium Proceedings (pp. 667-673). IEEE.

Cheung, Y.-N., Zhu, Y., Cheng, C.-H., Chao, C., & Leung, W. W.-F.,2008. A novel fluidic strain sensor for large strain measurement. Sensors and Actuators A: Physical, 147(2), 401–408. doi:10.1016/j.sna.2008.05.013

Chossat, J.-B., Park, Y.-L., Wood, R. J., & Duchaine, V.,2013. A Soft Strain Sensor Based on Ionic and Metal Liquids. IEEE Sensors Journal, 13(9), 3405–3414. doi:10.1109/jsen.2013.2263797

Cortes, C., & Vapnik, V., 1995. Machine Learning, 20(3), 273–297. doi:10.1023/a:1022627411411

De Deuge, M., Quadros, A., Hung, C. and Douillard, B., 2013, December. Unsupervised feature learning for classification of outdoor 3d scans. In Australasian Conference on Robitics and Automation (Vol. 2, p. 1).

El Haddad, M. H., Topper, T. H., & Smith, K. N., 1979. *Prediction of non propagating cracks. Engineering Fracture Mechanics, 11(3), 573–584.* doi:10.1016/0013-7944(79)90081-x

Fan, F.-R., Tian, Z.-Q., & Lin Wang, Z., 2012. Flexible triboelectric generator. Nano Energy, 1(2), 328–334. doi:10.1016/j.nanoen.2012.01.004

Farhat, N. H., 1989. Microwave diversity imaging and automated target identification based on models of neural networks. Proceedings of the IEEE, 77(5), 670–681. doi:10.1109/5.32058

Frutiger, A., Muth, J. T., Vogt, D. M., Mengüç, Y., Campo, A., Valentine, A. D., … Lewis, J. A.,2015. *Capacitive Soft Strain Sensors via Multicore-Shell Fiber Printing. Advanced Materials*, 27(15), 2440–2446. doi:10.1002/adma.201500072

Fu Jie Huang, & LeCun, Y., n.d. Large-scale Learning with SVM and Convolutional for Generic Object Categorisation. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06). doi:10.1109/cvpr.2006.164

Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep Learning. 1st ed. Cambridge, Massachusetts, United States: The MIT Press.

Hughes, J., Culha, U., Giardina, F., Guenther, F., Rosendo, A., & Iida, F.,2016. *Soft Manipulators and Grippers: A Review. Frontiers in Robotics and AI*, 3. doi:10.3389/frobt.2016.00069

Hunt, A., Freriks, M., Sasso, L., Esfahani, P. M., & HosseinNia, S. H., 2018. IPMC Kirigami: A Distributed Actuation Concept. 2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS). doi:10.1109/marss.2018.8481176

Hutton, D.V., 2004. *Fundamentals of finite element analysis*. McGraw-hill.
Hwang, D.-G., & Bartlett, M. D., 2018. *Tunable Mechanical Metamaterials through Hybrid Kirigami Structures. Scientific Reports, 8(1).* doi:10.1038/s41598-018-21479-7

J. H. Winters., 1988. "Superresolution for ultrasonic imaging in air using neural network," in Proc. Int. Conf Neural Networks, vol. 1, 1988, pp. 609-616.

Keulemans, G., Pelgrims, P., Bakula, M., Ceyssens, F., & Puers, R., 2014. An Ionic Liquid Based Strain Sensor for Large Displacements. Procedia Engineering, 87, 1123–1126. doi:10.1016/j.proeng.2014.11.362

Kier, W. M., & Stella, M. P.,2007. *The arrangement and function of octopus arm musculature and connective tissue. Journal of Morphology*, 268(10), 831. doi:10.1002/jmor.10548

Kim, J. H., Yoon, S. H., & Sohn, K. H., 1996. A robust boundary-based object recognition in occlusion environment by hybrid Hopfield neural networks. Pattern Recognition, 29(12), 2047–2060. doi:10.1016/s0031-3203(96)00043-x

Leguillon, D., Quesada, D., Putot, C., & Martin, E., 2007. *Prediction of crack initiation at blunt notches and cavities – size effects. Engineering Fracture Mechanics, 74(15), 2420–2436.* doi:10.1016/j.engfracmech.2006.11.008

Li, H., Wang, W., Yang, Y., Wang, Y., Li, P., Huang, J., Li, J., Lu, Y., Li, Z., Wang, Z., Fan, B., Fang, J., Song, W., 2020. *Kirigami-Based Highly Stretchable Thin Film Solar Cells that are Mechanically Stable for More than 1000 Cycles. ACS Nano.* doi:10.1021/acsnano.9b06562

Liao, Q., Mohr, M., Zhang, X., Zhang, Z., Zhang, Y., & Fecht, H.-J., 2013. Carbon fiber–ZnO nanowire hybrid structures for flexible and adaptable strain sensors. Nanoscale, 5(24), 12350. doi:10.1039/c3nr03536k

Lipomi, D. J., Vosgueritchian, M., Tee, B. C.-K., Hellstrom, S. L., Lee, J. A., Fox, C. H., & Bao, Z., 2011. Skin-like pressure and strain sensors based on transparent elastic films of carbon nanotubes. Nature Nanotechnology, 6(12), 788–792. doi:10.1038/nnano.2011.184

Liu, X., Mwangi, M., Li, X., O'Brien, M., & Whitesides, G. M., 2011. Paper-based piezoresistive MEMS sensors. Lab on a Chip, 11(13), 2189. doi:10.1039/c1lc20161a

Logan, D.L., 2011. *A first course in the finite element method.* Cengage Learning.

Madden, J. D. W., Vandesteeg, N. A., Anquetil, P. A., Madden, P. G. A., Takshi, A., Pytel, R. Z., … Hunter, I. W., 2004. Artificial Muscle Technology: Physical Principles and Naval Prospects. IEEE Journal of Oceanic Engineering, 29(3), 706–728. doi:10.1109/joe.2004.833135

Mattmann, C., Amft, O., Harms, H., Troster, G., and Clemens, F.,2007. "*Recognising upper body postures using textile strain sensors,*" in 11th IEEE International Symposium on Wearable Computers (Boston, MA: IEEE), 29–36.

Maturana, D., & Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/iros.2015.7353481

Nasrabadi, N. M., & Li, W., 1991. Object recognition by a Hopfield neural network. IEEE Transactions on Systems, Man, and Cybernetics, 21(6), 1523–1535. doi:10.1109/21.135694

Noguchi, Y., Sekitani, T., & Someya, T., 2006. Organic-transistor-based flexible pressure sensors using ink-jet-printed electrodes and gate dielectric layers. Applied Physics Letters, 89(25), 253507. doi:10.1063/1.2416001

Pham, G. T., Park, Y.-B., Liang, Z., Zhang, C., & Wang, B., 2008. Processing and modeling of conductive thermoplastic/carbon nanotube films for strain sensing. Composites Part B: Engineering, 39(1), 209–216. doi:10.1016/j.compositesb.2007.02.024

Prokhorov, D. V., 2009. Object recognition in 3D lidar data with recurrent neural network. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. doi:10.1109/cvprw.2009.5204114

Pugal, D., Kim, K. J., & Aabloo, A., 2011. An explicit physics-based model of ionic polymer-metal composite actuators. Journal of Applied Physics, 110(8), 084904. doi:10.1063/1.3650903

R, Muralidharan & Chandrasekar, Chandramouli., 2011. Object Recognition using SVM-KNN based on Geometric Moment Invariant. International Journal of Computer Trends and Technology. 1. 215 - 220.

Rafsanjani, A., Zhang, Y., Liu, B., Rubinstein, S. M., & Bertoldi, K., 2018. Kirigami skins make a simple soft actuator crawl. Science Robotics, 3(15), eaar7555. doi:10.1126/scirobotics.aar7555

Rogers, J. A., Bao, Z., Baldwin, K., Dodabalapur, A., Crone, B., Raju, V. R., … Drzaic, P., 2001. *Paper-like electronic displays: Large-area rubber-stamped plastic sheets of electronics and microencapsulated electrophoretic inks*. Proceedings of the National Academy of Sciences, 98(9), 4835–4840. doi:10.1073/pnas.091588098

Shaffer, M. S. P., Fan, X., & Windle, A. H., 1998. Dispersion and packing of carbon nanotubes. Carbon, 36(11), 1603–1612. doi: 10.1016/S0008-6223(98)00130-4

Shahinpoor, M., Bar-Cohen, Y., Simpson, J. O., & Smith, J., 1998. Ionic polymer-metal composites (IPMCs) as biomimetic sensors, actuators and artificial muscles - a review. Smart Materials and Structures, 7(6), R15–R30. doi:10.1088/0964-1726/7/6/001

Shyu, T. C., Damasceno, P. F., Dodd, P. M., Lamoureux, A., Xu, L., Shlian, M., … Kotov, N. A., 2015. *A kirigami approach to engineering elasticity in nanocomposites through patterned defects. Nature Materials, 14(8), 785–789.* doi:10.1038/nmat4327

Song, Z., Wang, X., Lv, C., An, Y., Liang, M., Ma, T., He, D., Zheng, Y., Huang, S., Yu, H., Jiang, H., 2015. *Kirigami-based stretchable lithium-ion batteries. Scientific Reports, 5(1).* doi:10.1038/srep10988

Suganuma, F., Shimamoto, A., & Tanaka, K.,1999. Development of a differential optical-fiber displacement sensor. Applied Optics, 38(7), 1103. doi:10.1364/ao.38.001103

Tognetti, A., Lorussi, F., Bartalesi, R., Quaglini, S., Tesconi, M., Zupone, G., & De Rossi, D., 2005. Journal of NeuroEngineering and Rehabilitation, 2(1), 8. doi:10.1186/1743-0003-2-8

Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I., 2020. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. Mathematics and Computers in Simulation, 177, 232–243. doi:10.1016/j.matcom.2020.04.031

Visin, Francesco, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron C. Courville and Yoshua Bengio., 2015. "ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks." ArXiv abs/1505.00393: pp. 1-3

Wang, Y., Yang, Y., & Wang, Z. L., 2017. Triboelectric nanogenerators as flexible power sources. Npj Flexible Electronics, 1(1). doi:10.1038/s41528-017-0007-8

Wang, Z. L., 2011. Self-Powered Nanosensors and Nanosystems. Advanced Materials, 24(2), 280–285. doi:10.1002/adma.201102958

Watanabe, S., & Yoneyama, M., 1992. *An ultrasonic visual sensor for three-dimensional object recognition using neural networks*. IEEE Transactions on Robotics and Automation, 8(2), 240–249. doi:10.1109/70.134277

Winstone, B., Griffiths, G., Pipe, T., Melhuish, C & Rossiter, J.,2013. *TACTIP - Tactile Fingertip Device, Texture Analysis Through Optical Tracking of Skin Features.* 323–333. doi:10.1007/978-3-642-39802-5

Xu, C., Zi, Y., Wang, A. C., Zou, H., Dai, Y., He, X., … Wang, Z. L., 2018. *On the Electron-Transfer Mechanism in the Contact-Electrification Effect.* Advanced Materials, 30(15), 1706790. doi:10.1002/adma.201706790

Yong-Lae Park, Bor-Rong Chen, & Wood, R. J.,2012. *Design and Fabrication of Soft Artificial Skin Using Embedded Microchannels and Liquid Conductors*. IEEE Sensors Journal, 12(8), 2711–2718. doi:10.1109/jsen.2012.2200790

Young, Scott, & Nasrabadi., 1994. Object recognition using multi-layer Hopfield neural network. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94. doi:10.1109/cvpr.1994.323860

Zaidi, N. A., & Squire, D. M., 2010. Local Adaptive SVM for Object Recognition. 2010 International Conference on Digital Image Computing: Techniques and Applications. doi:10.1109/dicta.2010.44

Zhou, J., Gu, Y., Fei, P., Mai, W., Gao, Y., Yang, R., … Wang, Z. L., 2008. Flexible Piezotronic Strain Sensor. Nano Letters, 8(9), 3035–3040. doi:10.1021/nl802367t

Zhu, M., Sun, Z., Zhang, Z., Shi, Q., He, T., Liu, H., … Lee, C. 2020. *Haptic-feedback smart glove as a creative human-machine interface (HMI) for virtual/augmented reality applications.* Science Advances, 6(19), eaaz8693. doi:10.1126/sciadv.aaz8693

APPENDICES

APPENDIX A: Bend Characterization Code


```
int sensor = A0; //Defining sensor pin
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  Serial.begin(9600);
  pinMode(sensor, OUTPUT);
}
void loop() {
        sensorValue = analogRead(sensor); //Read sensor value
        Serial.print("SensorValue= "); //print sensor value to serial
        Serial.println(sensorValue);
}
```

APPENDIX B1: Obtaining Training Samples (Python)

```python
import serial
import time
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
import csv

ser = serial.Serial('COM3', 9600)#timeout = 5
ser.flushInput()
time.sleep(3)
emp = []
val1 = []
DATASET = 150 #number of datapoints/set
FILE = "data_1_1_"
data = np.zeros(shape=(DATASET,5))
output = 0
filecount=0 #changing sample number
dataset_dict = {
    'shape': {
        0: 'Rectangular',
        1: 'Cylinder',
        2: 'Pyramid',
        3: 'Sphere',
    }
}

while True:
    while True: #determine when the button is pressed, send '1' when button pushed
        print("Push button to read data")
        startBit = ser.readline()
        if startBit == b'1\r\n':
            print("Button Pushed!")
            ser.write(b'1')
            break;

    for datapoint in range(DATASET+1):
        ser_bytes = ser.readline()
        if datapoint != 0: #ignore first read (bad data)
            #decode from bytes to str remove \r\n from end of line, and turn to list
            decode = ser_bytes.decode("utf-8")[:-2].split()
            #['sensorValue=', '142', 'sensorValue1=', '935', 'sensorValue2=', '199']
            count = 0
            for ind, val in enumerate(decode): #arrange values to array (row = trials, col= finger)
                if ind%2: #find odd no.
                    data[datapoint-1][count] = int(val)
                    count+=1
    ser.write(b'S')
    print("Stop sentinel sent!")
    filecount+=1
```

```
with open(FILE+str(filecount)+'.csv','w') as f: #save data
    np.savetxt(f, data, delimiter = ',',fmt='%i')

plt.plot(data)
plt.ylabel('bend angle')
plt.show()
```

APPENDIX C2: Obtaining Training Samples (Arduino code)

```
int thumb = A0; int index = A1; int middle = A2; int ring = A3; int pinky = A4;
int LED = 13;
int button = 7;
int sensorValue = 0; int sensorValue1 = 0; int sensorValue2 = 0; int sensorValue3 = 0; int
sensorValue4 = 0;
int buttonState = LOW;
int prevState = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 60;
int incomingByte = 0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(button);

  if ((millis() - lastDebounceTime) > debounceDelay){
    if(buttonState == HIGH && prevState == LOW){
      lastDebounceTime = millis();
      prevState = HIGH;
      Serial.println("1"); //send start sentinel to PC
    }
    else if(buttonState == LOW){
      prevState = LOW;
      lastDebounceTime = millis();
    }
  }

  if(Serial.available()>0){
    incomingByte = Serial.read();
    //Serial.println(incomingByte);

    while(Serial.available() == 0){
      if(incomingByte == 49){ //ASCII for 1
        sensorValue = analogRead(thumb);
        sensorValue1 = analogRead(index);
        sensorValue2 = analogRead(middle);
        sensorValue3 = analogRead(ring);
        sensorValue4 = analogRead(pinky);

        Serial.print("thumb= "); Serial.print(sensorValue);
        Serial.print(" ");
        Serial.print("index= "); Serial.print(sensorValue1);
        Serial.print(" ");
        Serial.print("middle= "); Serial.print(sensorValue2);
        Serial.print(" ");
```

```
Serial.print("ring= "); Serial.print(sensorValue3);
      Serial.print(" ");
      Serial.print("pinky= "); Serial.println(sensorValue4);
    }
    else if(incomingByte == 83){ //ASCII for 'S' : Stop sentinel
      incomingByte = 0;
      break;
    }
   }
  }
}
```

APPENDIX C: Real-time Prediction

```python
import tensorflow as tf
import serial
import time
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
import os
from os.path import isfile, join
import glob
import matplotlib.image as mpimg
from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix


DATASET = 150
dataset_dict = {
    'shape': {
        0: 'D:\Studies\FYP\FYP1.5\AI\DATA\CUBOID.png',
        1: 'D:\Studies\FYP\FYP1.5\AI\DATA\CYLINDER.png',
        2: 'D:\Studies\FYP\FYP1.5\AI\DATA\\NO_OBJ.png',
        3: 'D:\Studies\FYP\FYP1.5\AI\DATA\SPHERE.png',
    }
}


#serial comm
ser = serial.Serial('COM3', 9600)#timeout = 5
ser.flushInput()
time.sleep(3)

fileList = glob.glob("1_DATA COMPILED\*.csv")
x = []
y = []
#read train data
for fileName in fileList:
    with open(fileName, 'r') as f:
        data = f.readlines()
        #convert to float and list
        data = [list(map(float, i.strip().split(','))) for i in data]
    x.append(data)

    #get the label from filename and append it to a list
    y.append(int(fileName.split('_')[2]))
```

```python
x = np.array(x)
y = np.array(y)
x = x.reshape((800, 150*5))

#80% train(160 samples from each object), 20% test(40 samples from each object)
SEED = 51
X_train, X_test, Y_train, Y_test = train_test_split(x,y, test_size = 0.2, random_state = SEED,stratify = y)

#fitting data to SVM
clf = svm.SVC(kernel = 'rbf')#, probability = True)
clf.fit(X_train, Y_train)

y_pred = clf.predict(X_test)
#prediction accuracy
print("accuracy", metrics.accuracy_score(Y_test, y_pred))
print("Precision:",metrics.precision_score(Y_test, y_pred, average = None))
print("Recall:",metrics.recall_score(Y_test, y_pred, average = None))
plot_confusion_matrix(clf, X_test, Y_test)
plt.show()
def read_real_time_data(DATASET): #read data from glove
    data = np.zeros(shape=(DATASET,5))
    while True: #determine when the button is pressed, send '1' when button pushed
        print("Push button to read data")
        startBit = ser.readline()
        if startBit == b'1\r\n':
            print("Button Pushed!")
            ser.write(b'1')
            break;

    for datapoint in range(DATASET+1):
        ser_bytes = ser.readline()
        #print("Received: " + str(ser_bytes))
        if datapoint != 0: #ignore first read (bad data)
            #decode from bytes to str remove \r\n from end of line, and turn to list
            decode = ser_bytes.decode("utf-8")[:-2].split()
            #['sensorValue=', '142', 'sensorValue1=', '935', 'sensorValue2=', '199']
            count = 0
            for ind, val in enumerate(decode): #arrange values to array (row = trials, col= finger)
                if ind%2: #find odd no.
                    data[datapoint-1][count] = int(val)
                    count+=1
    ser.write(b'S')
    print("Stop sentinel sent!")

    data = np.array(data)
    return data
```

```python
while True:
    f, (ax1, ax2) = plt.subplots(1,2)
    x_rt = read_real_time_data(DATASET)
    x_rt1 = x_rt.reshape((1, 150*5))
    y_rt = clf.predict(x_rt1)
    print(y_rt)
    imgLoc = dataset_dict['shape'][int(y_rt)]
    print(imgLoc)
    Y_img = mpimg.imread(imgLoc)
    ax1.plot(x_rt)
    ax1.set_title("Raw Data")
    ax2.imshow(Y_img)
    ax2.set_title("Predicted Object")
    mng = plt.get_current_fig_manager()
    mng.window.state('zoomed')
    plt.show()
```