

الجامعة الإسلامية العالمية ماليزيا  
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA  
يونیورسitiٰ اسلام انتارا بعثینا ملیسیا

KULLIYAH OF INFORMATION AND  
COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

FYP PRELIMINARY REPORT

---

SIGN LANGUAGE RECOGNITION USING DEEP LEARNING

---

MHD KHALED MAEN

1523591

SUPERVISED BY

ASSOC. PROF. DR. AMELIA RITAHANI

DECEMBER 2018

SEMESTER 1, 2018 / 2019

# **DECLARATION**

I hear by declare that this report is the result of my own investigations, except where otherwise stated. I also clear that it has not been previously or currently submitted as a whole for any other degree at IIUM or other institutions.

MHD KHALED MAEN (1523591)

Signature: .....

Date: .....

# APPROVAL PAGE

I certified that I have supervise can read this study and that in my opinion, confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as final year project paper a partial fulfilment for a degree of bachelor of Computer Science (Honours).

Assoc. Prof. Dr. Amelia Ritahani (Supervisor)

Department of Computer Science

Kulliyyah of Information and Communication Technology

International Islamic University Malaysia

# ABSTRACT

Communication is an essential part of our life. Unfortunately, some of us were born in various types of disability such as deaf, since hearing impaired people cannot listen they loosed the ability to learn how to speak so they developed a new communication way to interact with other people by using distinct hand gestures, which was not enough to overcome this issue since most of hearing people do not understand sign language, even now with all technologies and tools it is remain a challenging problem to solve. For the mentioned reason, the intention of this paper is to improve the ordinary model to translate the sign language gestures into a voice. In that, Deep learning is remarkably serviceable for this mission, first by detecting a hand in a video frame using Convolutional Neural Network (CNN) algorithm followed by recognizing the letter and state the matching sound. The accuracy achieved for a hand gesture detection using CNN model MobileNet-SSD is more than 90 % for all the proposed signs.

## **ACKNOWLEDGEMENT**

This project has been completed with support from my supervisor, Assoc. Prof. Dr. Amelia Ritahani, many thanks for her wonderful collaboration and consultation sessions. Furthermore, I would like to thank my coordinators, Asst. Prof. Dr. Hamwira Yaacob and Asst. Prof. Dr. Norzariyah Binti Yahya for their assistance helping me and others by organizing weekly sessions towards writing perfect report step-by-step. Last but not least, I want to thank my awesome parents for their support and time helping me reach the end of my undergraduate study, without them, I could not achieve that.

# Table of Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introduction</b>                     | <b>1</b> |
| 1.1      | Background . . . . .                    | 1        |
| 1.2      | Problem Statement . . . . .             | 2        |
| 1.3      | Objectives . . . . .                    | 2        |
| 1.4      | Scope . . . . .                         | 2        |
| 1.5      | Significance . . . . .                  | 3        |
| 1.6      | Timeline . . . . .                      | 3        |
| <b>2</b> | <b>Literature review</b>                | <b>4</b> |
| 2.1      | Sign Language . . . . .                 | 4        |
| 2.2      | Deep Learning . . . . .                 | 6        |
| 2.2.1    | Activation Function . . . . .           | 7        |
| 2.2.2    | Weights . . . . .                       | 9        |
| 2.2.3    | Bias . . . . .                          | 9        |
| 2.3      | Convolutional Neural Networks . . . . . | 9        |
| 2.3.1    | Feature Learning . . . . .              | 10       |
| 2.3.2    | Classification . . . . .                | 12       |
| 2.4      | Previous works . . . . .                | 12       |

|          |   |           |
|----------|---|-----------|
| 2.5      | Summary . . . . .                             | 19        |
| <b>3</b> | <b>Methodology</b>                            | <b>20</b> |
| 3.1      | Environment . . . . .                         | 21        |
| 3.2      | Data set . . . . .                            | 21        |
| 3.3      | Image processing . . . . .                    | 23        |
| 3.3.1    | Image annotation . . . . .                    | 23        |
| 3.3.2    | XML to Comma Separated Values (CSV) . . . . . | 24        |
| 3.4      | TensorFlow Object Detection API . . . . .     | 24        |
| 3.5      | algorithm . . . . .                           | 24        |
| 3.6      | Hand detection . . . . .                      | 25        |
| 3.6.1    | Single-Shot Detector (SSD) . . . . .          | 25        |
| 3.7      | Voice producing . . . . .                     | 26        |
| 3.8      | Tools . . . . .                               | 27        |
|          | <b>References</b>                             | <b>29</b> |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Sign Language Alphabet. Retrieved from www.startASL.com . . . . .          | 5  |
| 2.2  | Typical deep learning model. Retrieved from www.medium.com . . . . .       | 6  |
| 2.3  | Sigmoid Function . . . . .   | 7  |
| 2.4  | Tanh Function . . . . .  | 8  |
| 2.5  | ReLU Function . . . . .  | 8  |
| 2.6  | Max pooling with 2x2 filter and stride = 2. Retrieved: Wikipedia . . . . . | 11 |
| 2.7  | Complete CNN architecture. Retrieved: medium.com . . . . .                 | 12 |
| 2.8  | Architecture of the proposed deep CNN . . . . .                            | 14 |
| 2.9  | Proposed Deep CNN architecture . . . . .                                   | 15 |
| 2.10 | VGG16 architecture. Retrieved from www.cs.toronto.edu . . . . .            | 16 |
| 2.11 | AlexNet architecture. Retrieved from www.saagie.com . . . . .              | 16 |
| 2.12 | Classification results of modified networks . . . . .                      | 17 |
| 2.13 | parallel convolutional neural network . . . . .                            | 18 |
| 3.1  | System block diagram . . . . .   | 20 |
| 3.2  | The letter K in American Sign language . . . . .                           | 22 |
| 3.3  | XML file for letter A . . . . .  | 23 |
| 3.4  | CSV file for image test data . . . . .                                     | 24 |

|  |    |
|--|----|
| 3.5 MobileNet-SSD Retrieved from <a href="https://hey-yahei.cn/2018/08/08/MobileNets-SSD/">https://hey-yahei.cn/2018/08/08/MobileNets-SSD/</a> | 25 |
| 3.6 SSD. Retrieved from <a href="https://www.semanticscholar.org">https://www.semanticscholar.org</a>  | 27 |

# **List of Tables**

|  |    |
|--|----|
| 2.1 Summary of the literature review . . . . . | 19 |
|--|----|

# **Chapter 1**

## **Introduction**

### **1.1 Background**

Communication is a process of sending and receiving data among individuals. People communicate with each other's by a considerable measure of ways yet the best way is eye to eye correspondence. Numerous individuals trust that the significance of communication is like the importance of breathing. Indeed, communication facilitates the spread of knowledge and structures connections between individuals.

Deep learning added an immense lift to the already rapidly developing field of computer vision. With deep learning, a lot of new utilization of computer vision techniques have been presented and they are currently ending up some portion of our regular day to day existence.

Alongside with the intensity of the present computers, there are now various algorithms that were developed to empower the computers to perform tasks such as object tracking and pattern recognition.

In this study, the attention will be on hand gestures detection and make an interpretation of them into voice.

## **1.2 Problem Statement**

Communication difficulties arising from damage to hearing directly have an effect on the standard of life. Difficulties in communication could end in deviations within the emotional and social development which will have a major impact on the standard of lifetime of every one. It is well recognized that hearing is crucial to speech and language development, communication, and learning. People with listening difficulties due to hearing loss or auditory processing problems continue to be an under-identified and under-served population. The earlier the matter is known and intervention began, the less serious the ultimate impact (Frajtag1 & Jelinic2, 2017).

The communication between hearing-impaired and other individuals is a colossal gap need to be filled up. In order to overcome this challenge many researches and products have been developed to solve this problem, but there is a lot to be enhanced.

## **1.3 Objectives**

- To collect sign language gestures dataset.
- To develop a hand gesture into voice algorithm.
- To construct a hand gesture into voice model.
- To evaluate and test the constructed model.

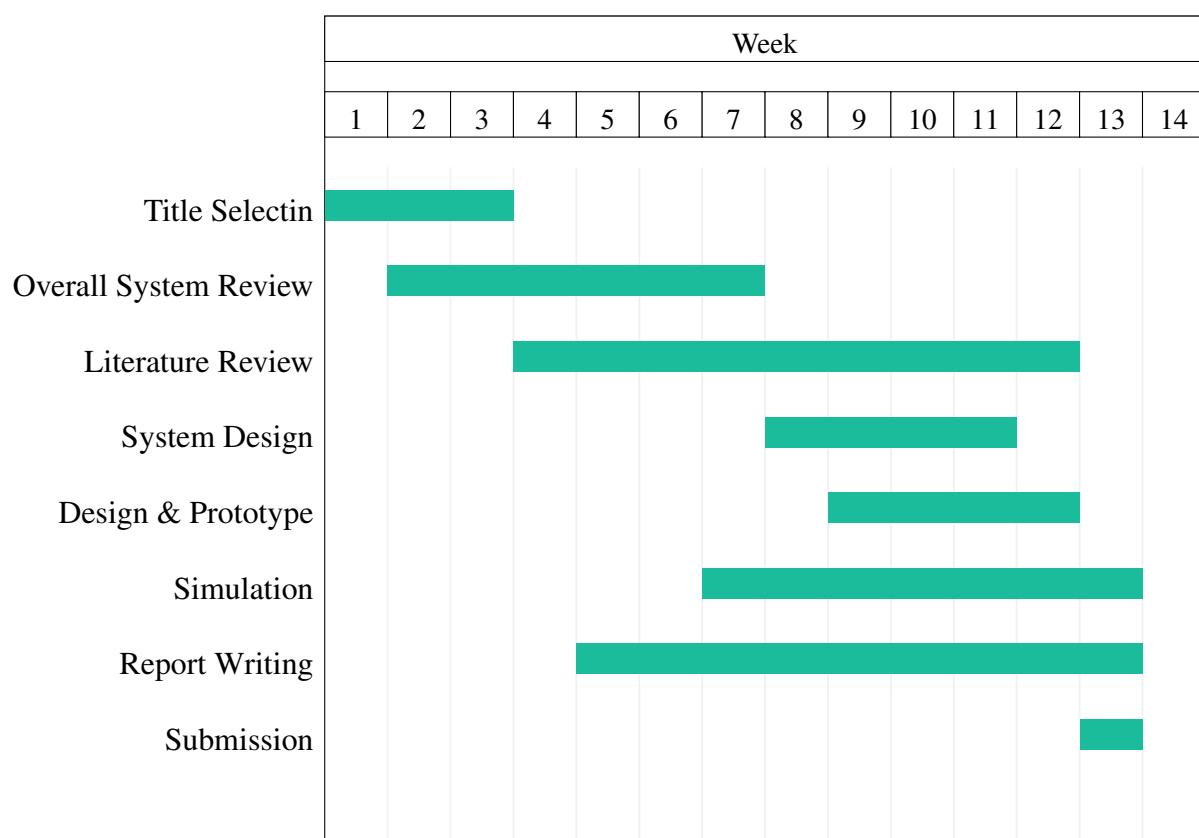
## **1.4 Scope**

This research aims to develop a sign language recognition algorithm, and converting it into voice.

## 1.5 Significance

Help the hearing-impaired community to communicate with hearing ones, in order to make a strong connected community.

## 1.6 Timeline



# **Chapter 2**

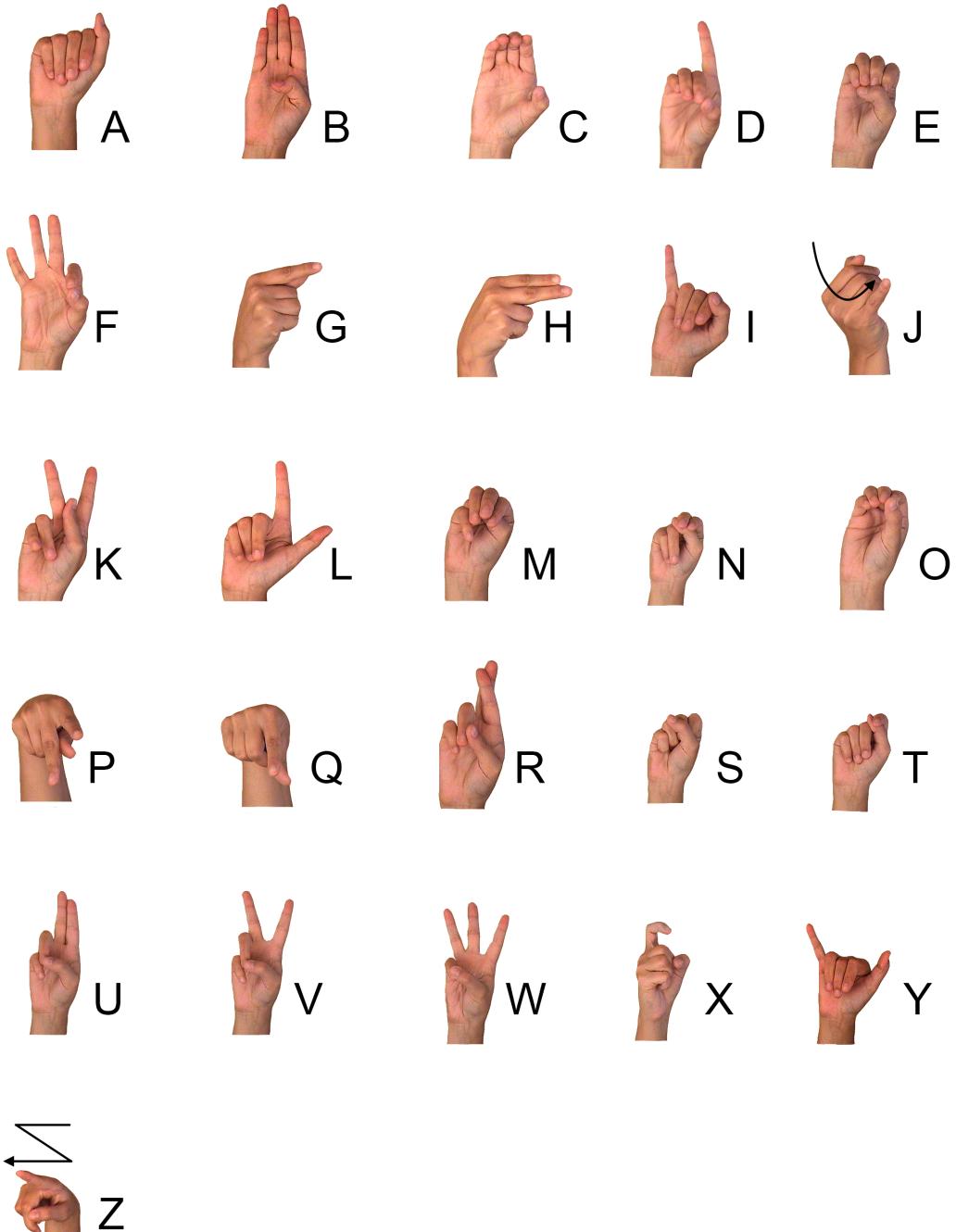
## **Literature review**

This chapter includes reviews of other previous researcher and their proposed methods they used in implementing deep learning to recognize hand gestures. These researches will help to grasp the knowledge to achieve the project's objectives.

### **2.1 Sign Language**

In spoken languages words being pronounced by vocal cords which produce sound wave can be heard, however for deaf people this sounds are not heard (Ishizaka & Flanagan, 1972), so sign language is based on the visual tools to communicate and receive information. American Sign Language (ASL) is completely different from English. It has the basic features of a language such as the letters as shown in 2.1.

## American Sign Language Alphabet



Copyright © 2008 StartASL.com

Figure 2.1: Sign Language Alphabet. Retrieved from [www.startASL.com](http://www.startASL.com)

## 2.2 Deep Learning

Deep learning is a machine learning subfield that deals with algorithms based on the structure and function of the brain called artificial neural networks. In other words, it mirrors the brain's functioning. Deep learning algorithms are similar to the structure of the nervous system in which each neuron connects and passes information. Deep learning models work in the layers and three layers of a typical model at least 2.2. Each layer accepts and passes the information from the previous layer to the next layer.

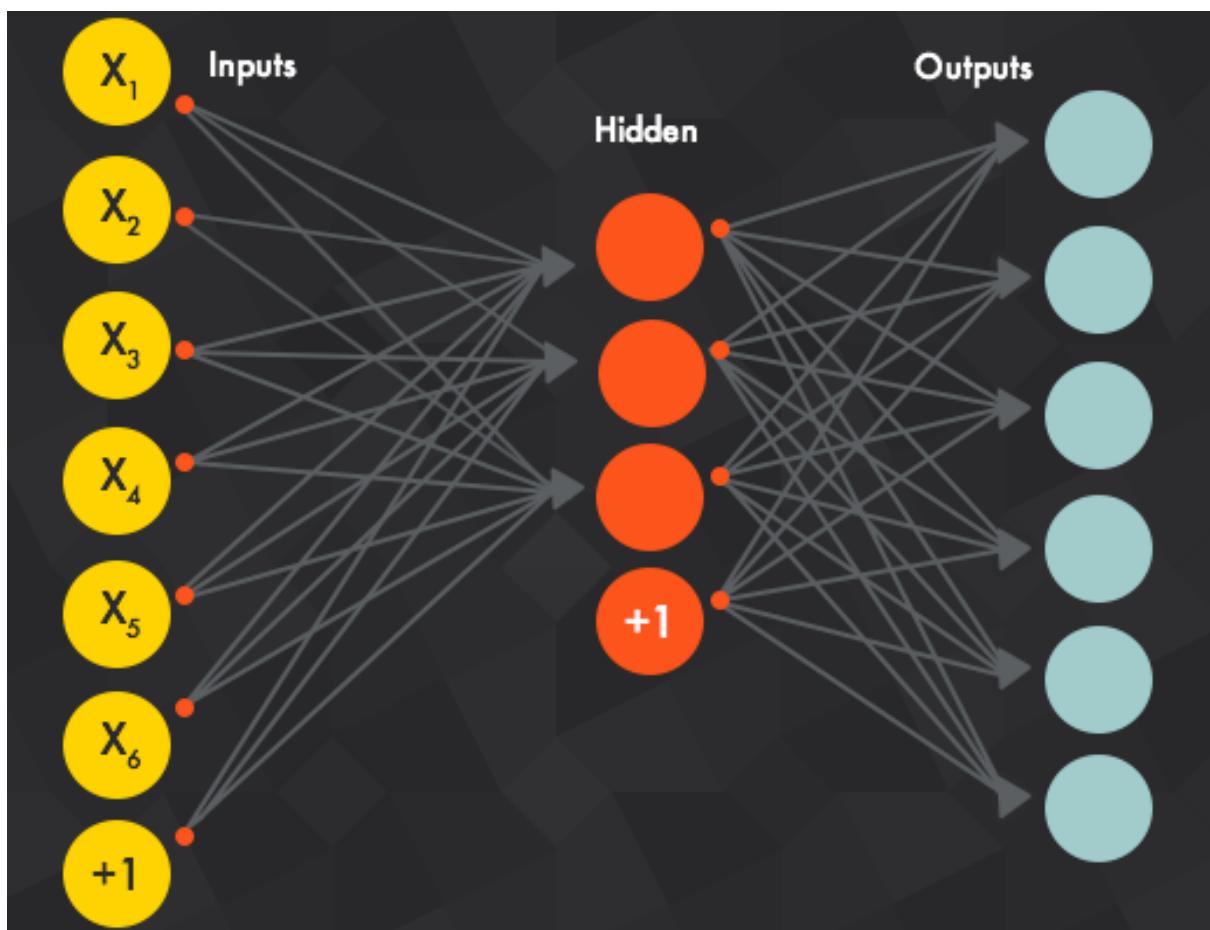


Figure 2.2: Typical deep learning model. Retrieved from [www.medium.com](http://www.medium.com)

Deep learning models tend to perform well with quantity of data while old machine learning models do not improve after a saturation.

One of differences between machine learning and deep learning model is on the feature extraction area. Feature extraction is done by human in machine learning whereas deep learning model figure out by itself.

### 2.2.1 Activation Function

Activation functions are functions that decide what the output of the node should be given the inputs into the node? Since the activation function determines the actual output, the outputs of a layer are often referred to as "activations." The most well known activation functions are Sigmoid 2.3, Tanh 2.4 and Rectified Linear Unit ReLU 2.5.

#### Sigmoid Function:

Non linear activation function with output in range (0,1).

$$A = \frac{1}{1 + e^{-x}}$$

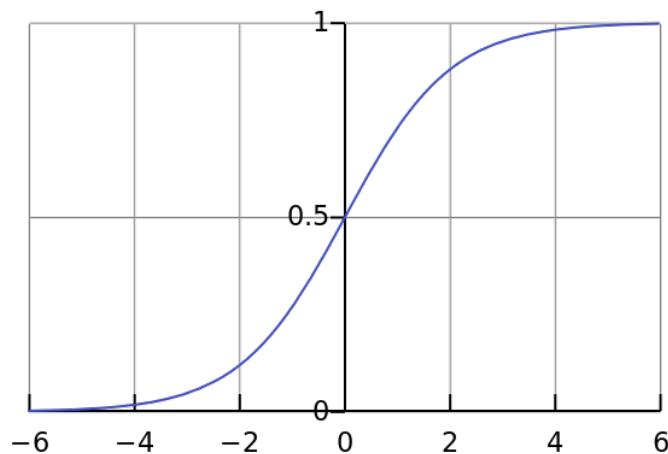


Figure 2.3: Sigmoid Function

### Tanh Function:

it is a scaled sigmoid function, bound to range (-1, 1).

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

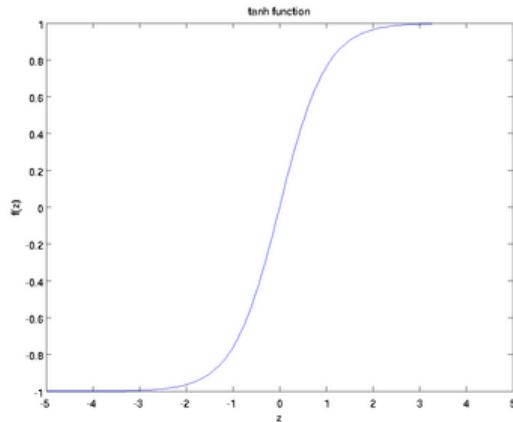


Figure 2.4: Tanh Function

### ReLU Function:

less computationally expensive than tanh and sigmoid, it is range (-1, infinity).

$$A(x) = \max(0, x)$$

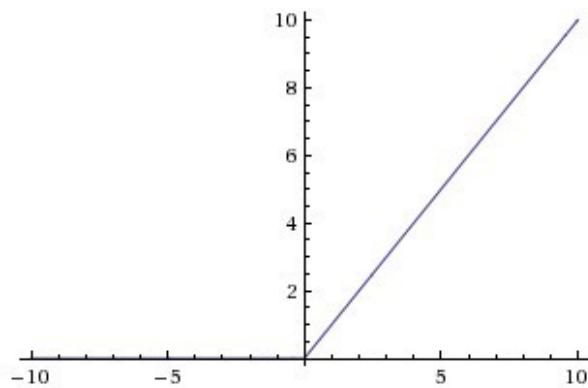


Figure 2.5: ReLU Function

### **2.2.2 Weights**

When input data enters a neuron, it is multiplied by a weight value which is assigned to the input. The neuron above the university example, for example, has two inputs, test scores and grades, so it has two related weights that can be individually adjusted. These weights start as random values and as the neural network learns more about what type of input data, the weights are adjusted based on any categorization errors resulting from previous weights it can be associated as  $m$  in the original linear equation.

$$y = mx + b$$

### **2.2.3 Bias**

Same as weights, biases are the learnable parameters of the deep learning model. The bias represents  $b$  in the original linear equation.

$$y = mx + b$$

## **2.3 Convolutional Neural Networks**

Convolutional Neural Networks are very similar to ordinary Neural Networks, they are made up of neurons that have learnable weights and biases. The architectures of ConvNet make the explicit assumption that the inputs are images which will be encoded to certain properties in the architecture. This makes the forward function more efficient to implement and reduces the number of parameters in the network considerably.

Convolutional Neural networks allow computers to see, meaning that Convnets is used to recognize images by transforming the original image into a class scoring through layers. CNN

was inspired by the visual cortex. Every time we see something, a series of layers of neurons gets activated, and each layer will detect a set of features such as lines, edges. The high level of layers will detect more complex features in order to recognize what we saw.

ConvNet has two parts: feature learning (Conv, Relu, and Pool) and classification (FC and softmax).

### 2.3.1 Feature Learning

#### CONV layer:

The objective of a Conv layer is to extract features of the input image. A part of the image is connected to the next Conv layer because if all the input pixels are connected to the Conv layer, it is too expensive to compute. Therefore we will apply dot products in all dimensions between a receptive field and a filter. The result of this operation is a single volume integer (feature map). Then we slide the filter through a Stride over the next receiving field of the same input image and recalculate the dot products between the new receiving field and the same filter. This process is repeated until we pass the entire input image. The output is the input on the next layer.

Some of the word that used interchangeably:

- Filter (Kernel): a small matrix used to detect features.
- Feature Map: the output volume formed by sliding the filter over the image and computing the dot product.
- Receptive field: a local region of the input image that has the same size as the Kernel.
- Depth: the number of filters.

- Stride: has the objective of producing smaller output volumes spatially.
- Padding: the process of adding extra pixels over around the image.

### ReLU layer:

Turning negative values into zeros. it has nothing to do with size of image and there are no hyperparameters.

### Pool Layer:

Reduce the dimension of the input, the computational complexity of the model and controls the overfitting. There are different types of pooling such as Average pooling, Max pooling and L2-norm pooling. However, the Max pooling is the most use, which takes the most important part (the pixel with the highest value)2.6.

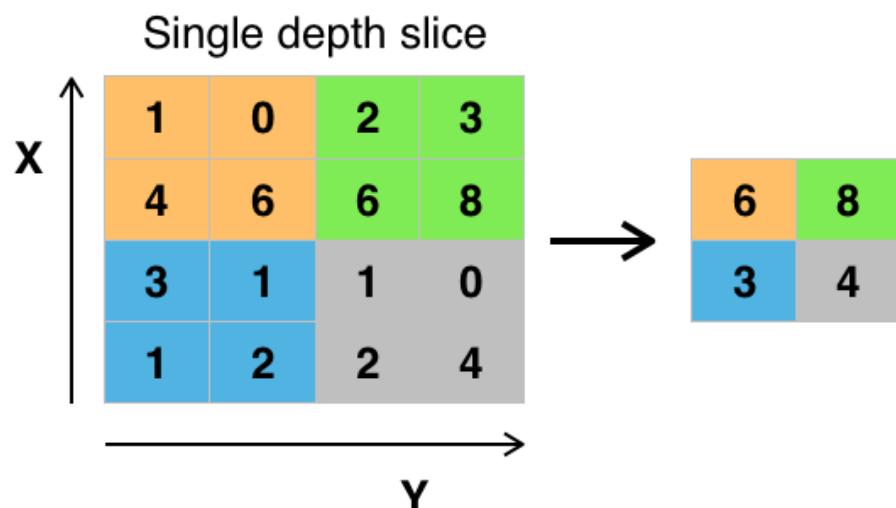


Figure 2.6: Max pooling with 2x2 filter and stride = 2. Retrieved: Wikipedia

### 2.3.2 Classification

#### Fully Connected Layer(FC):

Fully connected layers connect each neuron in a layer to every neuron in another layer.

The last connected layer uses softmax activation function.

#### Softmax:

Activation function generate features input image into many classes based on the dataset.

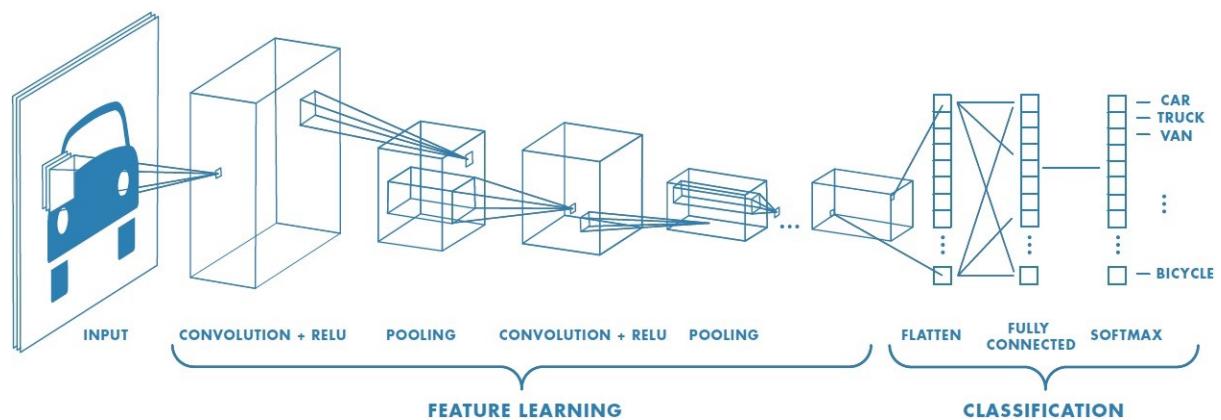


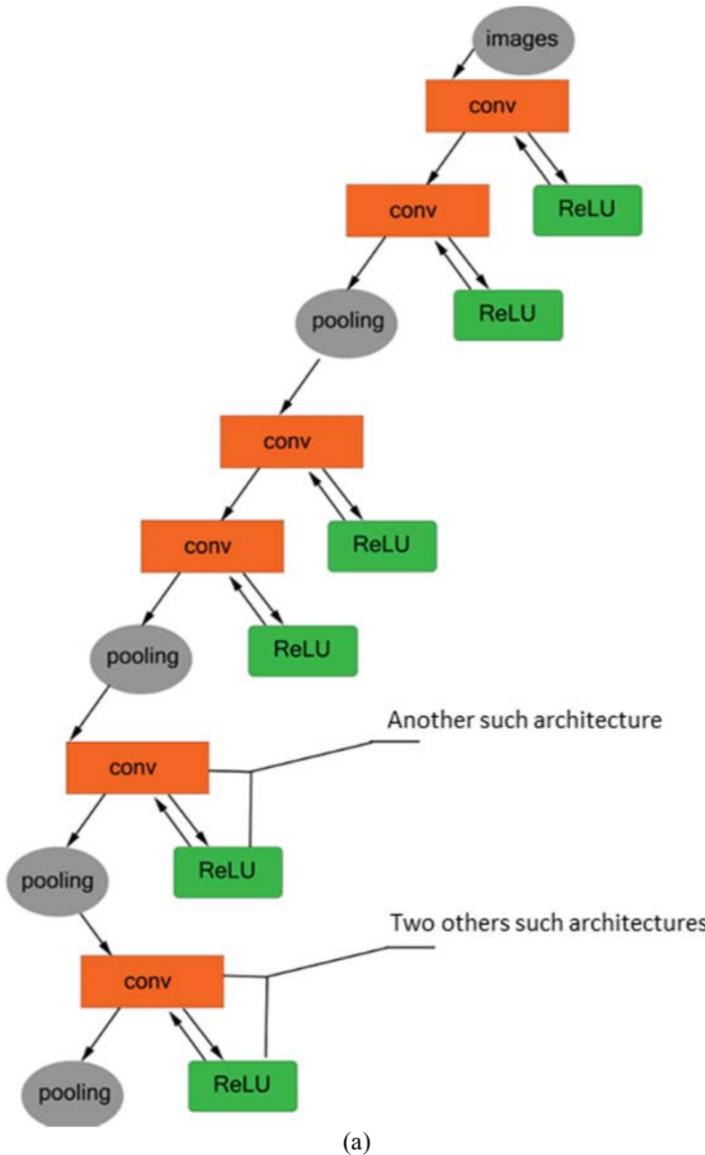
Figure 2.7: Complete CNN architecture. Retrieved: medium.com

## 2.4 Previous works

Following the same path many previous attempts has been conducted to find a solution for this problem, marvelous work has been done, but most of them was just a prove of concept were the authors concentrate on a few signs to be detected.

(Bao, Maqueda, del Blanco, & García, 2017), proposed a Deep convolutional neural network algorithm for hand-gesture recognition without hand localisation, since the hands only occupy about 10% of the image. They used a combination of 9 convolution layers, 3 fully

connected layers, interlaced with ReLU(Rectified Linear Unit) and dropout layers as shown in figure 2.8. Alongside this architecture the apply some image processing techniques to have sufficient computation efficiency and memory requirement. According to the paper the accuracy achieved was 97.1% in the images with simple backgrounds and 85.3% in the images with complex backgrounds. However, the main disadvantage of the proposed algorithm is the training set which only includes 7 different gestures, and it tends to have bad accuracy with complex backgrounds.



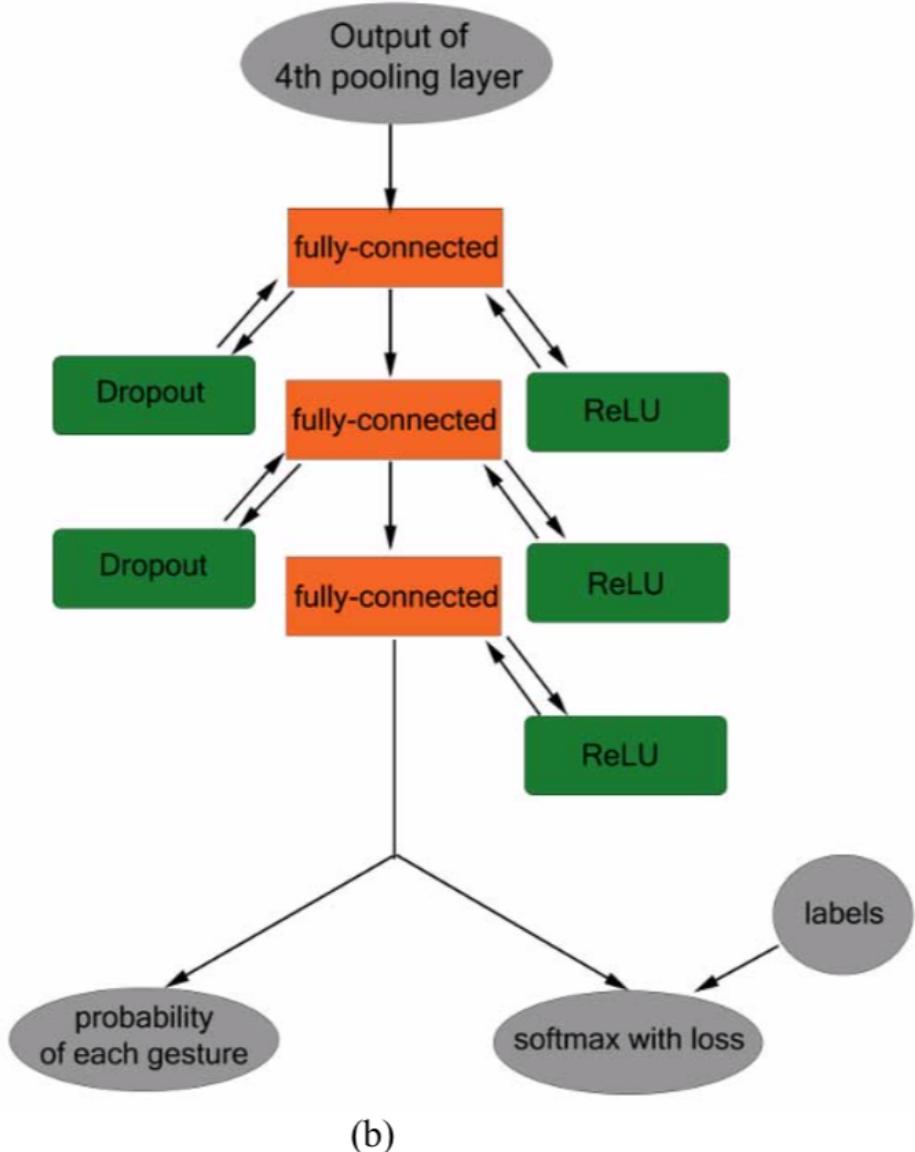


Figure 2.8: Architecture of the proposed deep CNN

(Rao, Syamala, Kishore, & Sastry, 2018), proposed a CNN architecture for classifying selfie sign language gestures. The CNN architecture is designed with four convolutional layers. Each convolutional layer with different filtering window sizes as shown in figure 2.9 They had a dataset with five different subjects performing 200 signs in 5 different viewing angles under various background environments. Each sign occupied for 60 frames or images in a video. The proposed model performed training on 3 batches to test the robustness of different training mode using caffe deep learning framework. However, the result accuracy was 92.88% need more training and improvements.

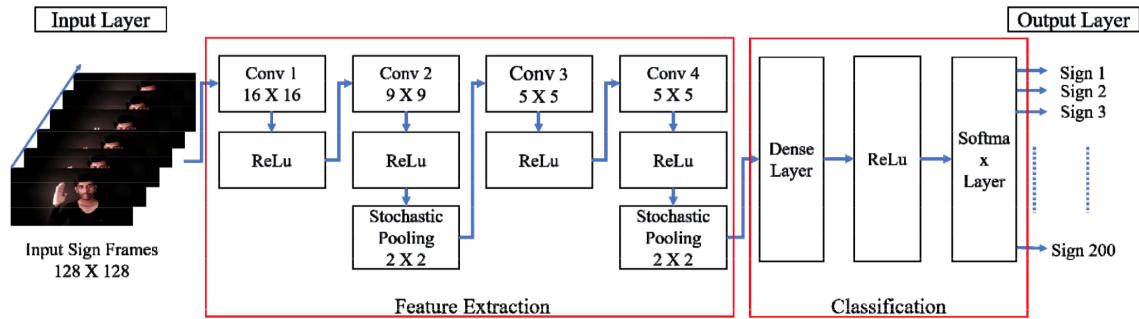


Figure 2.9: Proposed Deep CNN architecture

(Hussain, Saxena, Han, Khan, & Shin, 2017), introduced a CNN based classifier trained through the process of transfer learning over a pretrained convolutional neural network which is trained on a large dataset. We are using VGG16 figure 2.10 as the pretrained model. The According to the paper the accuracy was 93.09%, while using AlexNet figure 2.11 was 76.96%. the same problem here with the other papers which is the small number of sign that begin trained on 7 signs, and the accuracy need to be improved as well.

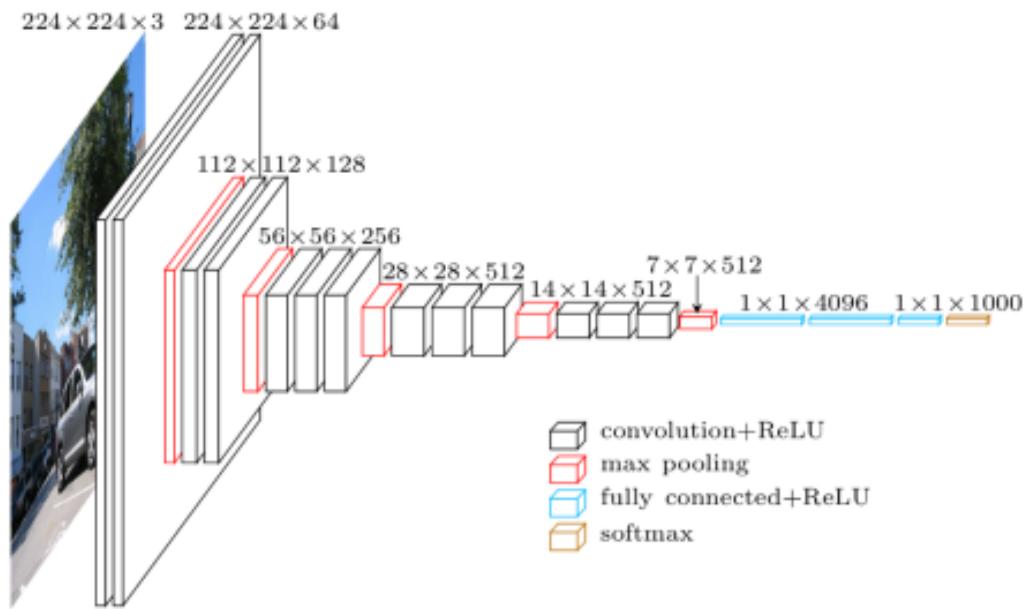


Figure 2.10: VGG16 architecture. Retrieved from [www.cs.toronto.edu](http://www.cs.toronto.edu)

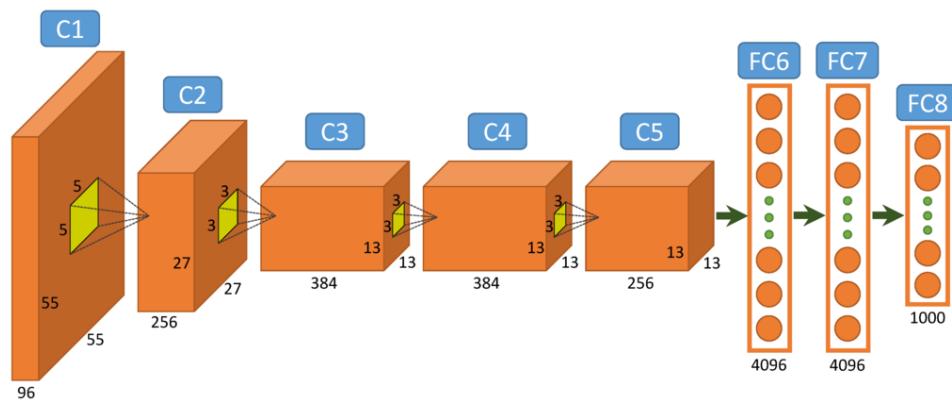


Figure 2.11: AlexNet architecture. Retrieved from [www.saagie.com](http://www.saagie.com)

(Pyo, Ji, You, & Kuc, 2016), introduced a depth-based hand data with convolution neural networks (CNNs). The hand gesture dataset has roughly 6,000 RGB-D images in each of 12 labels. In all, there are approximately 60,000 training images, 1S,000 validation images, and 12,000 training images. each time they were increasing the number of layers and testing the accuracy 2.12. They came with result that more number of layers, does not guarantee the increase of accuracy.

|                     | All folded | All stretched | Index thumb folded | Index thumb stretched | Middle index stretched | Middle index thumb stretched | Middle ring folded | Only index folded | Only little folded | Only ring folded | Only thumb folded | Only thumb stretched | Total  |
|---------------------|------------|---------------|--------------------|-----------------------|------------------------|------------------------------|--------------------|-------------------|--------------------|------------------|-------------------|----------------------|--------|
| <b>3conv 1fully</b> | 100.00%    | 99.47%        | 97.87%             | 100.00%               | 99.47%                 | 98.40%                       | 99.47%             | 54.79%            | 89.42%             | 99.04%           | 97.12%            | 96.15%               | 94.27% |
| <b>3conv 2fully</b> | 100.00%    | 100.00%       | 95.21%             | 100.00%               | 100.00%                | 97.87%                       | 99.47%             | 64.89%            | 97.12%             | 98.08%           | 100.00%           | 94.23%               | 95.57% |
| <b>3conv 3fully</b> | 100.00%    | 100.00%       | 97.87%             | 100.00%               | 100.00%                | 97.87%                       | 98.94%             | 62.23%            | 90.38%             | 97.12%           | 99.04%            | 91.35%               | 94.57% |
| <b>4conv 1fully</b> | 100.00%    | 92.55%        | 93.62%             | 98.94%                | 99.47%                 | 98.40%                       | 97.87%             | 55.32%            | 95.19%             | 98.08%           | 98.08%            | 100.00%              | 93.96% |
| <b>4conv 2fully</b> | 100.00%    | 96.81%        | 99.47%             | 100.00%               | 100.00%                | 98.40%                       | 99.47%             | 64.36%            | 91.35%             | 98.08%           | 100.00%           | 91.35%               | 94.94% |
| <b>4conv 3fully</b> | 100.00%    | 95.21%        | 98.94%             | 98.94%                | 99.47%                 | 96.28%                       | 97.87%             | 60.64%            | 92.31%             | 96.15%           | 99.04%            | 95.19%               | 94.17% |
| <b>5conv 1fully</b> | 99.47%     | 92.55%        | 96.81%             | 97.87%                | 99.47%                 | 97.87%                       | 97.87%             | 58.51%            | 94.23%             | 97.12%           | 98.08%            | 94.23%               | 93.67% |
| <b>5conv 2fully</b> | 100.00%    | 97.87%        | 98.94%             | 100.00%               | 100.00%                | 98.40%                       | 97.34%             | 53.19%            | 88.46%             | 97.12%           | 99.04%            | 94.23%               | 93.72% |
| <b>5conv 3fully</b> | 99.47%     | 94.68%        | 96.28%             | 99.47%                | 98.94%                 | 98.40%                       | 97.34%             | 52.66%            | 93.27%             | 95.19%           | 97.12%            | 91.35%               | 92.85% |
| <b>6conv 1fully</b> | 99.47%     | 94.68%        | 98.40%             | 97.34%                | 98.94%                 | 97.34%                       | 98.40%             | 55.32%            | 84.62%             | 98.08%           | 98.08%            | 95.19%               | 92.99% |
| <b>6conv 2fully</b> | 100.00%    | 97.34%        | 96.81%             | 97.87%                | 98.40%                 | 97.87%                       | 98.40%             | 53.72%            | 92.31%             | 95.19%           | 98.08%            | 94.23%               | 93.35% |
| <b>6conv 3fully</b> | 98.40%     | 43.09%        | 89.36%             | 97.34%                | 94.68%                 | 93.09%                       | 96.81%             | 47.87%            | 73.08%             | 88.46%           | 99.04%            | 93.27%               | 84.54% |

Figure 2.12: Classification results of modified networks

(Devineau, Moutarde, Xi, & Yang, 2018), introduced a 3D hand gesture recognition approach based on a deep learning model using Convolutional Neural Network (CNN). The proposed model only uses hand-skeletal data and no depth image. The model produced by multi-channel convolutional neural network with two feature extraction modules and a residual branch per channel 2.13. The achieved accuracy was a 91.28% classification accuracy for the 14 gesture classes case and an 84.35% classification accuracy for the 28 gesture classes case.

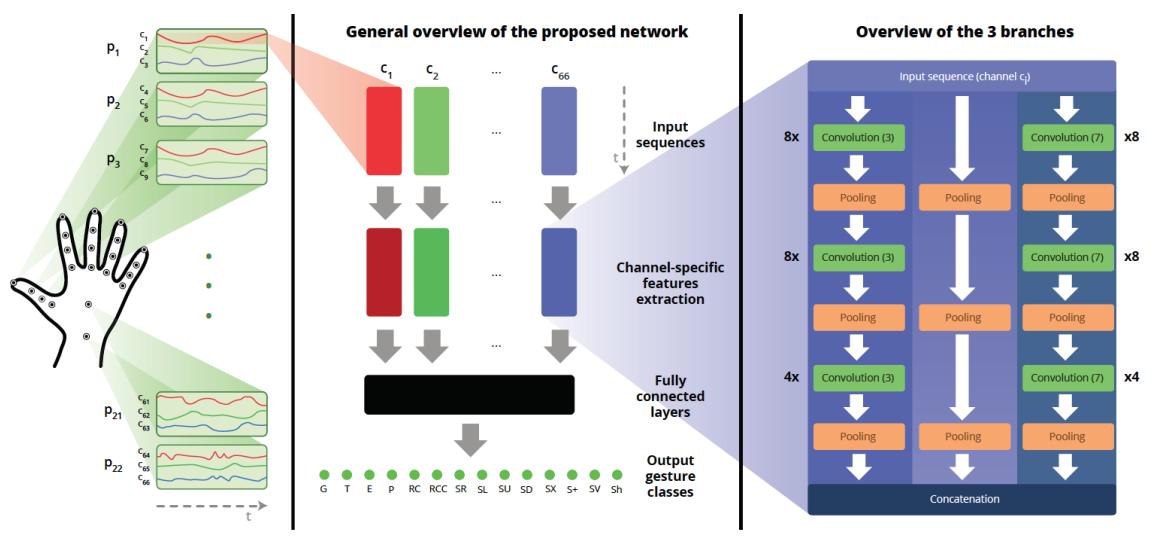


Figure 2.13: parallel convolutional neural network

## 2.5 Summary

This chapter illustrated some works have been done previously on hand gesture and sign language recognition using Machine Vision and deep learning (Convolutional Neural Network).

Table 2.1 shows the summary of the literature review.

Table 2.1: Summary of the literature review

| Title   | Year | Accuracy | Software  |
|---|------|----------|-----------|
| Tiny Hand Gesture Recognition without Localization via a Deep Convolutional Network | 2017 | 97.1%    | CNN       |
| Deep Convolutional Neural Networks for Sign Language Recognition                    | 2018 | 92.88%   | CNN       |
| Hand Gesture Recognition Using Deep Learning  | 2017 | 93.09%   | CNN VGG16 |
| Depth-based Hand Gesture Recognition using Convolutional Neural Networks            | 2016 | 95.57%   | CNN       |
| Deep Learning for Hand Gesture Recognition on Skeletal Data                         | 2018 | 91.28%   | MC-DCNN   |

# Chapter 3

## Methodology

Image recognition, voice producing, system design block diagram figure 3.1 and the flowchart of the research is presented in details alongside with the tools and algorithms in this chapter.

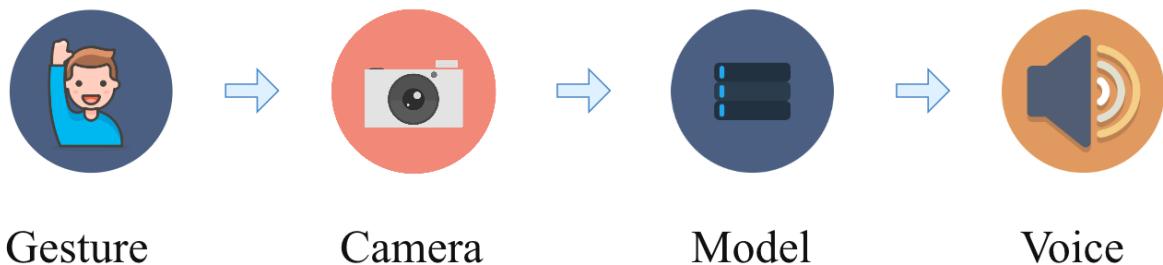


Figure 3.1: System block diagram

In order to get a model with satisfying results, we setup an environment with a powerful computational resources and large dataset.

### 3.1 Environment

Google Cloud Platform (GCP) is a cloud computing service provided by Google which provides a lot of products including cloud computing. Google Compute Engine (GCE) is the Infrastructure as a Service (IaaS) enables users to setup Virtual Machines (VMs) can be accessed via the developer console, command line interface (CLI) and RESTful API (Cloud, n.d.). The VM instance of choice run on an Ubuntu operating system version 16.04. Alongside, the VM instance is powered by 8 Graphics Processing Units (GPUs) of type NVIDIA Tesla K80 which engineered to boost throughput in real-world applications by 5-10x times, compared to Central Processing Unit (CPU).

### 3.2 Data set

The data set is a collection of images for American Sign Language (ASL) alphabet, contained of 87,000 images of 200x200 pixels frame size as shown in Figure 3. There are 29 classes, the first 26 classes are the letters A to Z and the last three classes are space delete and empty signs (ASL-Alphabet, n.d.).



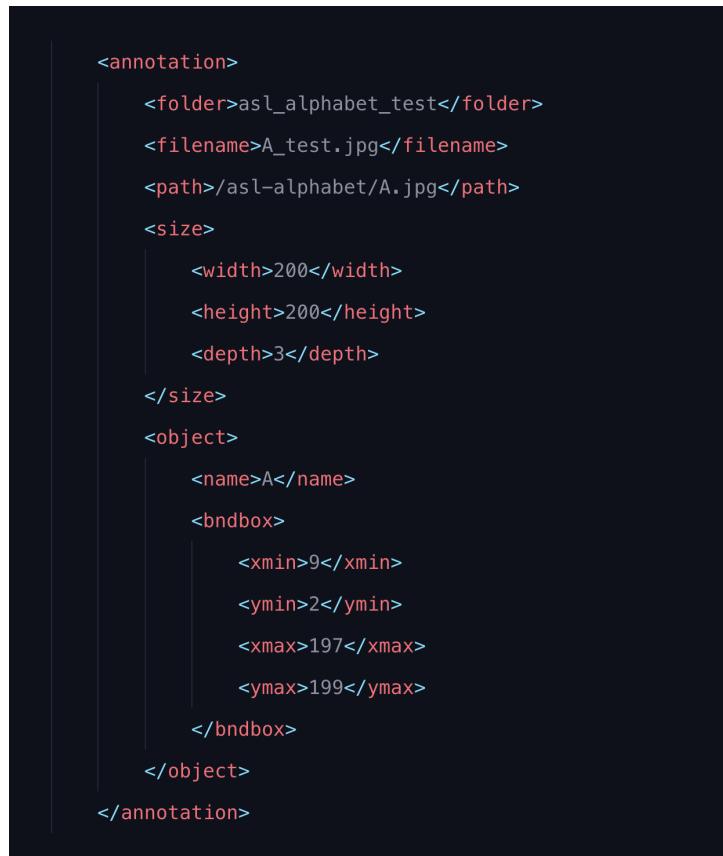
Figure 3.2: The letter K in American Sign language

### 3.3 Image processing

In order to extract useful information from an image, some methods need to be implemented:

#### 3.3.1 Image annotation

The process of annotating and labeling every object in the image in this case the sign has been labeled, so the machine can identify the important part of the image which has the object and the characteristics of it. The data of annotated image saved in eXtensible Mark-up Language (XML) file such as object class and the bounding box coordination Figure 3.3.



```
<annotation>
    <folder>asl_alphabet_test</folder>
    <filename>A_test.jpg</filename>
    <path>/asl-alphabet/A.jpg</path>
    <size>
        <width>200</width>
        <height>200</height>
        <depth>3</depth>
    </size>
    <object>
        <name>A</name>
        <bndbox>
            <xmin>9</xmin>
            <ymin>2</ymin>
            <xmax>197</xmax>
            <ymax>199</ymax>
        </bndbox>
    </object>
</annotation>
```

Figure 3.3: XML file for letter A

### 3.3.2 XML to Comma Separated Values (CSV)

The goal of this process is to convert the semi-structured data stored in the XML files into structured data stored in one file CSV so it can be transferred into TensorFlow(TF)Records Figure 3.4.

```
filename,width,height,class,xmin,ymin,xmax,ymax
F_test.jpg,200,200,F,15,10,170,150
G_test.jpg,200,200,G,17,21,120,170
L_test.jpg,200,200,L,20,10,137,165
M_test.jpg,200,200,M,12,32,180,200
```

Figure 3.4: CSV file for image test data

## 3.4 TensorFlow Object Detection API

Open source framework built on top of TensorFlow in order to help researchers to construct, train and deploy object detection models (Object-Detection-API, n.d.).

## 3.5 algorithm

The algorithm used in this experiment is MobileNet-SSD as shown in Figure 3.5; MobileNet is a base network which use depth-wise separable convolutions to build light weight deep neural networks provide high level features for classification or detection, after removing the fully connected layer at the end of the network and replace it with Single Shot Multibox Detector (SSD) one of the most popular algorithm in object detections (Liu et al., 2016). SSD generates a box with score of presence of each object category and adjusts the box to better

match the object shape. In addition SSD eliminates proposal generation and feature resampling stage by encapsulates all computation in a single network, alongside with the ability to combines predictions from multiple features maps with different resolutions to naturally handle objects of various sizes.

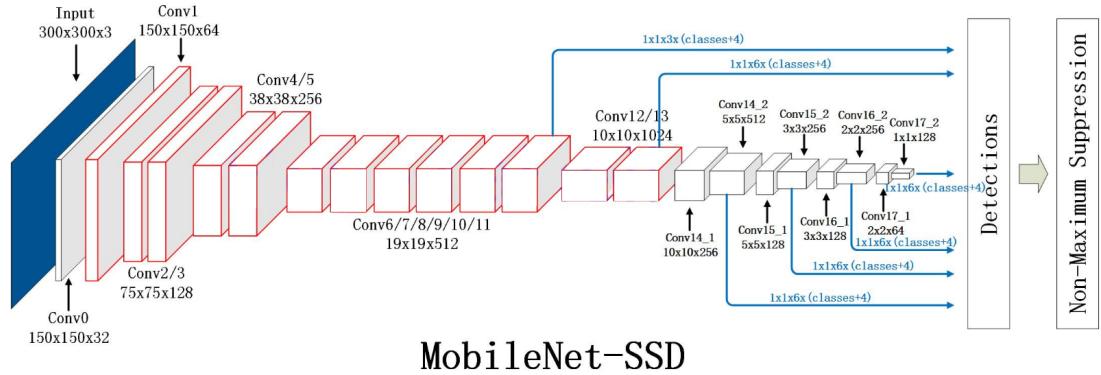


Figure 3.5: MobileNet-SSD Retrieved from <https://hey-yahei.cn/2018/08/08/MobileNets-SSD/>

## 3.6 Hand detection

The problem of hand recognition that hand occupied usually less than 25 percent of the image. To overcome this issue the model should be provided with high accurate detection algorithm, Right now there are so many good algorithms for object detection which can be utilize to detect a human hand.

### 3.6.1 Single-Shot Detector (SSD)

Unlike Faster R-CNN which perform regional proposals and region classifications in two steps. SSD does the two in a "single shot" jointly predict the bounding box and the class while it processes the image.

how it's work?

- Generate a set of feature maps with different scales by passing the image through sequence of convolutional layers (10x10, 6x6, 3x3 ...).
- Use a 3\*3 convolutional filter to evaluate bounding boxes for each location of the feature maps.
- predict bounding box of set and the class probability all together.
- The best predicted box called as "positive" label, alongside with the boxes that have IoU<sup>1</sup> value > 0.5

Sense SSD skip filtering step, it generates multiple bounding box with multiple shapes and most of them are negative example.

To fix this issue, SSD does two extra methods. First, non-maximum suppression:<sup>2</sup> to group overlapping boxes into one box by keeping the highest confidence Then, hard negative mining: to balance classes during the training process; subset the negative examples with the highest training loss with a 3:1 ratio of negatives for positives.(Liu et al., 2016)

### 3.7 Voice producing

After processing the image the CNN algorithm classify the gesture that presented in the image, the corresponding text (word, char, number) will be generated as voice that Simulate the human voice.

---

<sup>1</sup>Intersection over Union

<sup>2</sup>Object detection methods often output multiple detections which fully or partly cover the same object in an image.

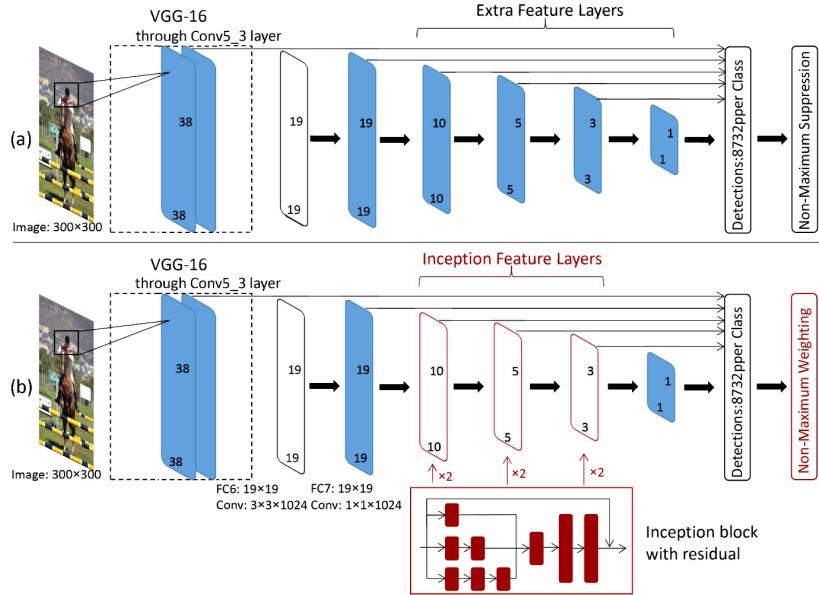


Figure 3.6: SSD. Retrieved from <https://www.semanticscholar.org>

## 3.8 Tools

The programming language in use is Python<sup>3</sup> along side with many libraries such as TensorFlow<sup>4</sup>, Keras<sup>5</sup>, OpenCV<sup>6</sup>, NumPy<sup>7</sup>, Pandas<sup>8</sup> and Matplotlib<sup>9</sup>. The model is being trained by

<sup>3</sup>Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991. <https://www.python.org/>

<sup>4</sup>TensorFlow is an open-source software library for dataflow programming across a range of tasks. <https://www.tensorflow.org/>

<sup>5</sup>Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. <https://keras.io/>

<sup>6</sup>OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. <https://opencv.org/>

<sup>7</sup>NumPy is the fundamental package for scientific computing with Python. <http://www.numpy.org/>

<sup>8</sup>Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. <https://pandas.pydata.org/>

<sup>9</sup>Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy

formats and interactive environments across platforms. <https://matplotlib.org/>

using Google Cloud Computing<sup>10</sup> service with Ubuntu as operating system.

---

<sup>10</sup>Google Compute Engine delivers virtual machines running in Google's innovative data centers and worldwide fiber network. <https://cloud.google.com/>

# References

- ASL-Alphabet, K. . (n.d.). *Kaggle — asl-alphabet*. Retrieved from <https://www.kaggle.com/grassknotted/asl-alphabet>
- Bao, P., Maqueda, A. I., del Blanco, C. R., & García, N. (2017, August). Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Transactions on Consumer Electronics*, 63(3), 251–257. doi: 10.1109/TCE.2017.014971
- Cloud, C. E. I. . C. E. . G. (n.d.). *Compute engine - iaas — compute engine — google cloud*. Retrieved from <https://cloud.google.com/compute/>
- Devineau, G., Moutarde, F., Xi, W., & Yang, J. (2018, May). Deep learning for hand gesture recognition on skeletal data. In *Proc. 13th ieee int. conf. automatic face gesture recognition (fg 2018)* (pp. 106–113). doi: 10.1109/FG.2018.00025
- Frajtag1, J. B., & Jelinic2, J. D. (2017). Communication problems and quality of life people with hearing loss. *Glob J Otolaryngol.*
- Hussain, S., Saxena, R., Han, X., Khan, J. A., & Shin, H. (2017, November). Hand gesture recognition using deep learning. In *Proc. int. soc design conf. (isocc)* (pp. 48–49). doi: 10.1109/ISOCC.2017.8368821

Ishizaka, K., & Flanagan, J. L. (1972). Synthesis of voiced sounds from a two-mass model of the vocal cords. *Bell system technical journal*, 51(6), 1233–1268.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37).

Object-Detection-API, T. . (n.d.). *Tensorflow — object-detection-api*. Retrieved from [https://github.com/tensorflow/models/tree/master/research/object\\_detection](https://github.com/tensorflow/models/tree/master/research/object_detection)

Pyo, J., Ji, S., You, S., & Kuc, T. (2016). Depth-based hand gesture recognition using convolutional neural networks. In *Ubiquitous robots and ambient intelligence (urai), 2016 13th international conference on* (pp. 225–227).

Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018, January). Deep convolutional neural networks for sign language recognition. In *Proc. conf. signal processing and communication engineering systems (spaces)* (pp. 194–197). doi: 10.1109/SPACES.2018.8316344