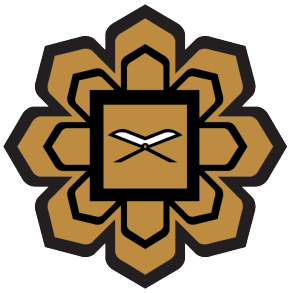


الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيبرسيتي إسلام انتارابغسا مليسيا

KULLIYYAH 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

List of Figures

List of Tables

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. Folks 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 gab 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 study sign language gestures.
- To develop a new hand gesture into voice algorithm.
- To construct a hand gesture into voice model.

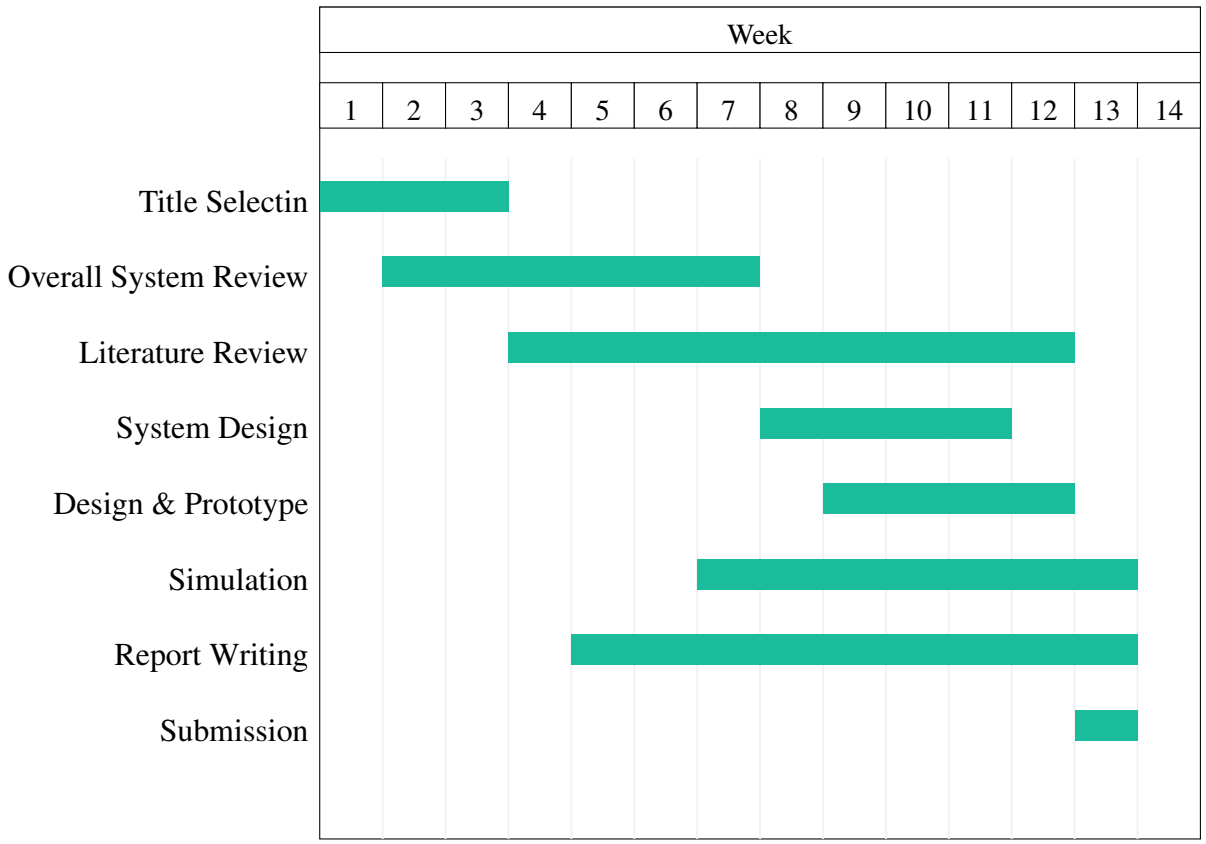
1.4 Scope

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 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 2.1. Each layer accepts and passes the information from the previous layer to the next layer.

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

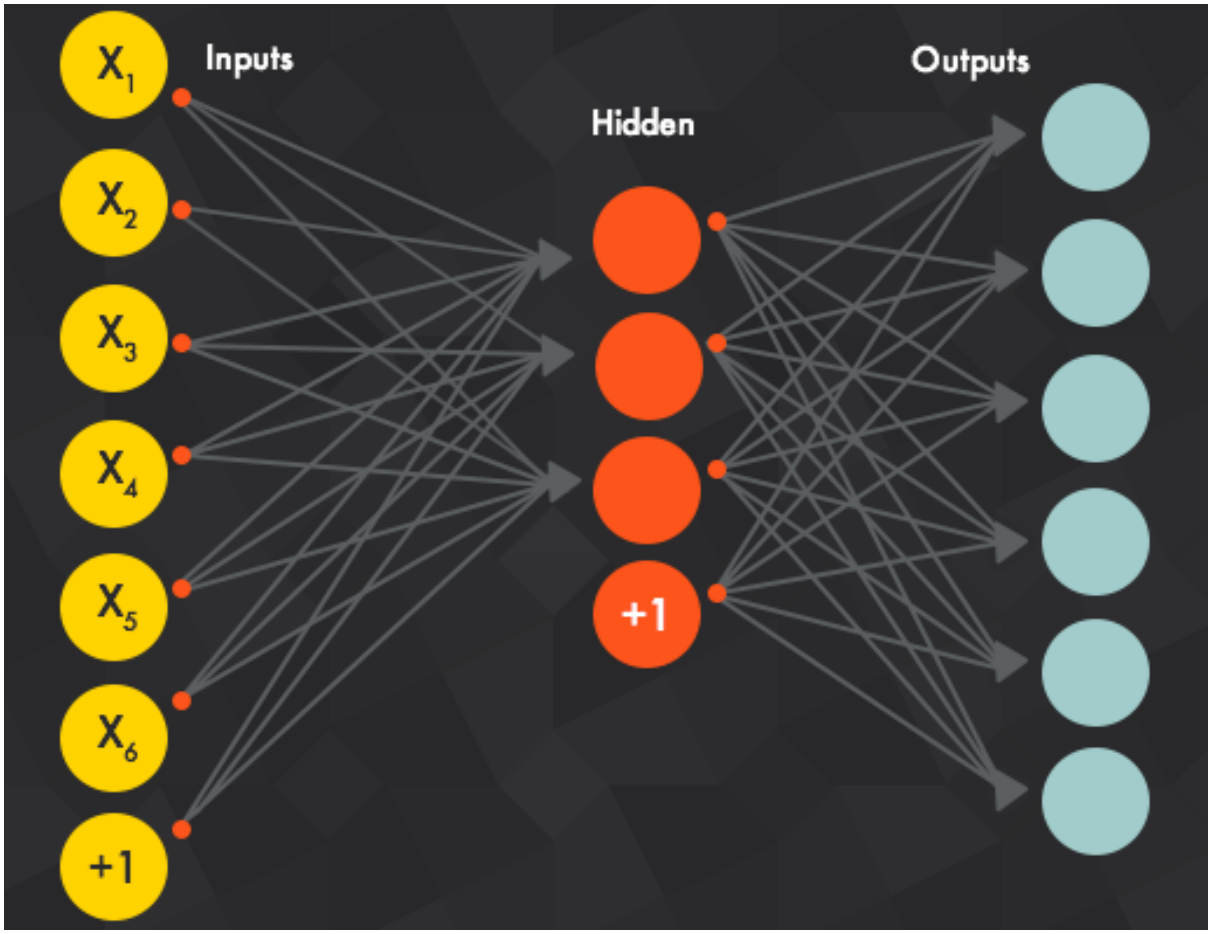


Figure 2.1: Typical deep learning model. Retrieved from www.medium.com

model figure out by itself.

2.1.1 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 2.2, Tanh 2.3 and Rectified Linear Unit ReLU 2.4.

Sigmoid Function:

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

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

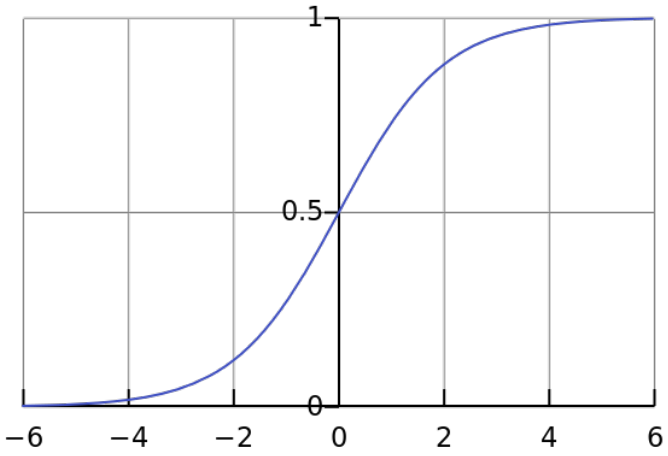


Figure 2.2: Sigmoid Function

Tanh Function:

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

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

ReLU Function:

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

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

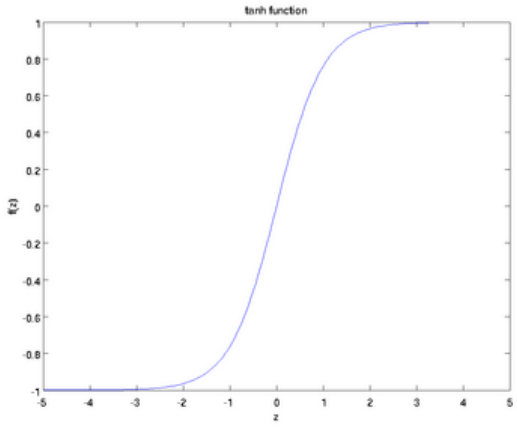


Figure 2.3: Tanh Function

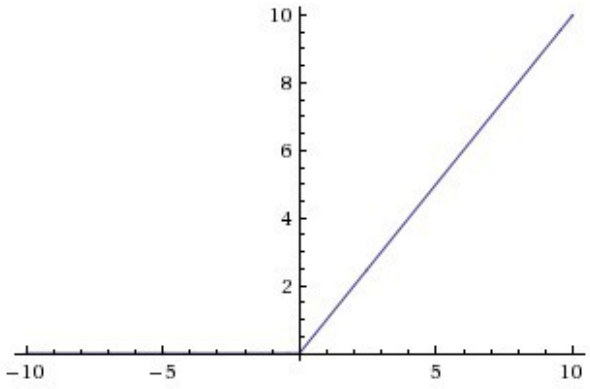


Figure 2.4: ReLU Function

2.1.2 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.

$$y = mx + b$$

2.1.3 Bias

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

$$y = mx + b$$

2.2 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).

2.2.1 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 (future 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 vule)2.5.

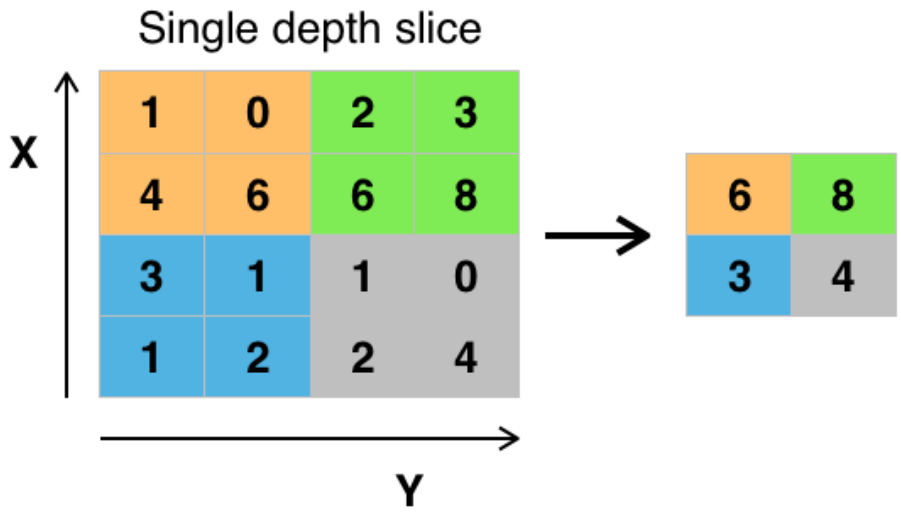


Figure 2.5: Max pooling with 2x2 filter and stride = 2. Retrieved: Wikipedia

2.2.2 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.

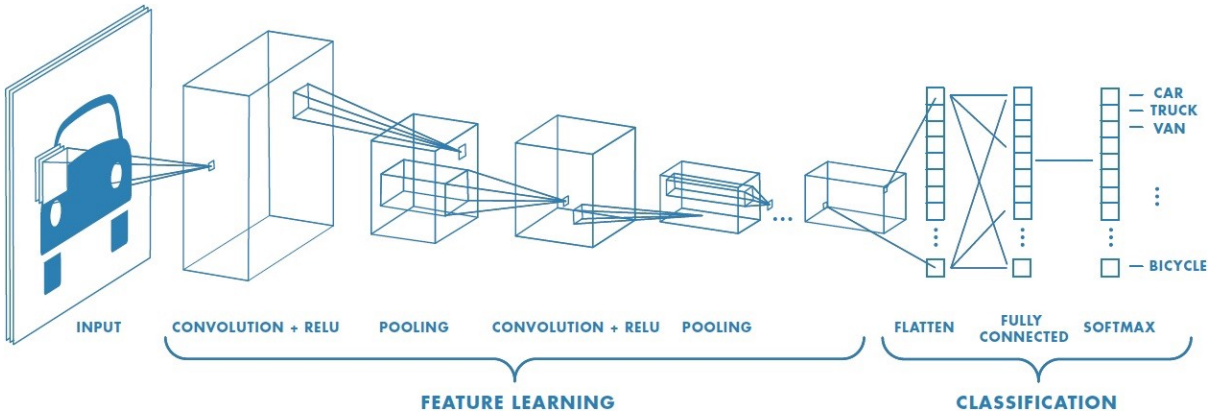
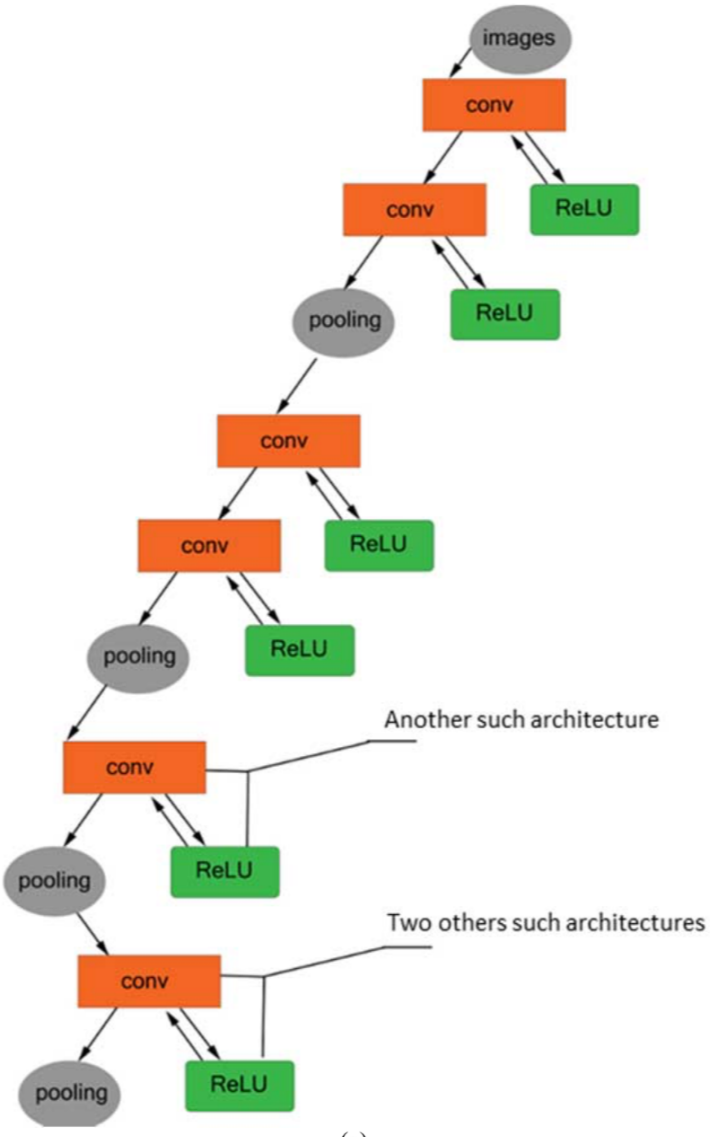


Figure 2.6: Complete CNN architecture. Retrieved: medium.com

2.3 Previous works

(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 2.7. 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 of the proposed algorithm is the training set which only includes 7 different gestures, and it tends to have bad accuracy with complex backgrounds.


images
conv
conv
ReLU
pooling
ReLU
conv
conv
ReLU
pooling
ReLU
Another such architecture
conv
pooling
ReLU
Two others such architectures
conv
pooling
ReLU

(a)

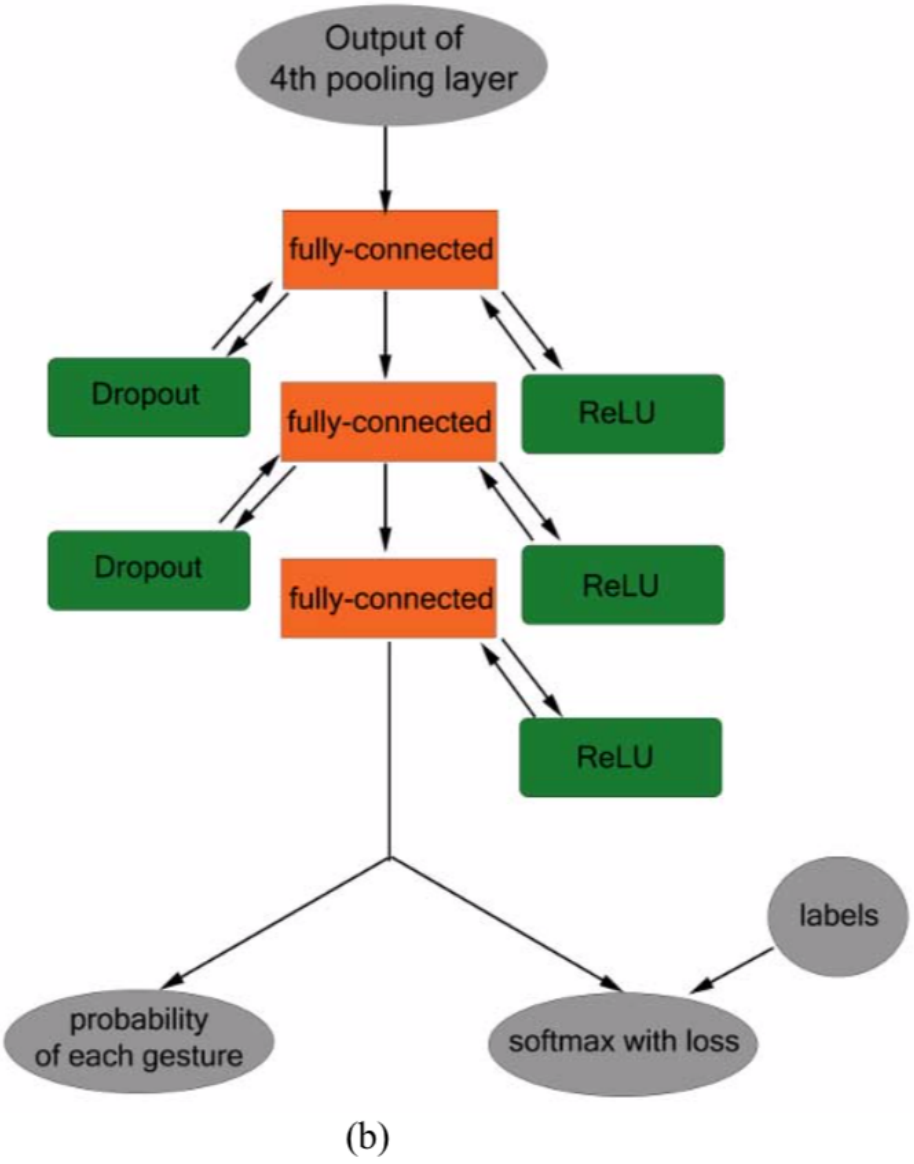


Figure 2.7: Architecture of the proposed deep CNN

(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 2.8 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.

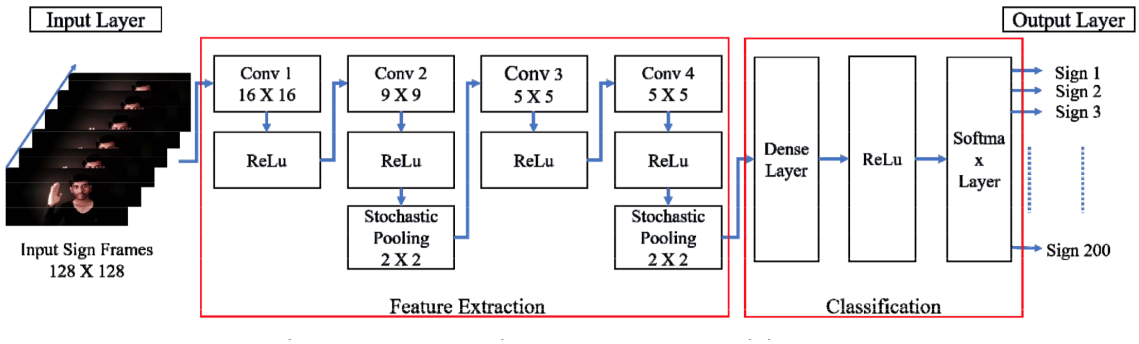


Figure 2.8: Proposed Deep CNN architecture

(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 2.9 as the pretrained model. The According to the paper the accuracy was 93.09%,while using AlexNet figure 2.10 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.

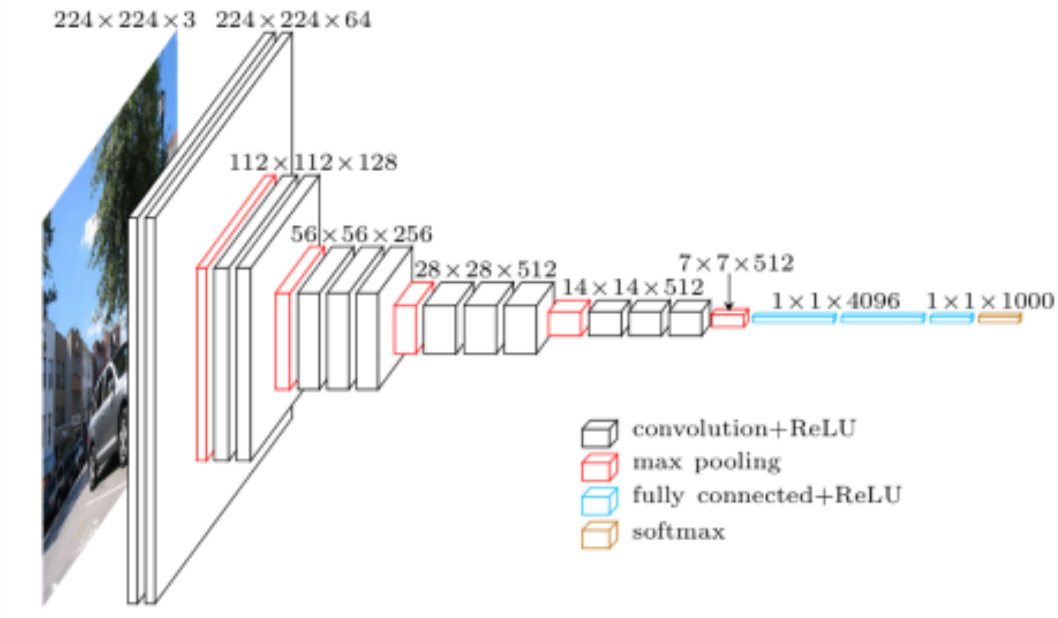


Figure 2.9: VGG16 architecture. Retrieved from www.cs.toronto.edu

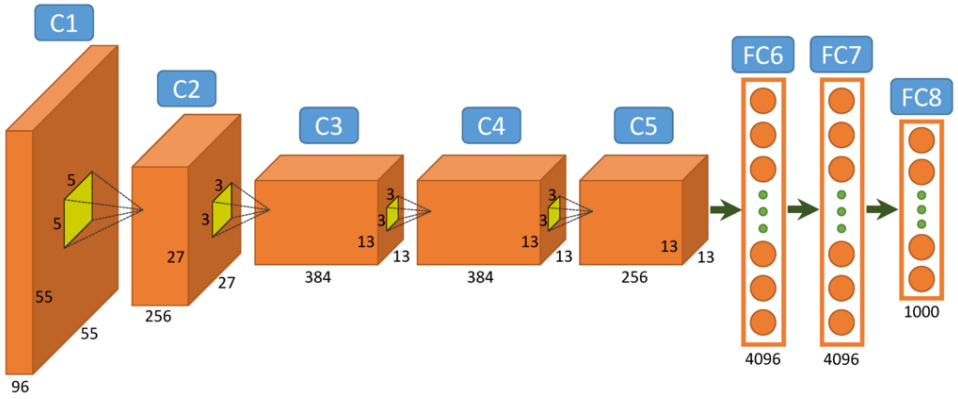


Figure 2.10: AlexNet architecture. Retrieved from www.saagie.com

(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 2.11. 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
3conv 1fully	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%
3conv 2fully	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%
3conv 3fully	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%
4conv 1fully	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%
4conv 2fully	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%
4conv 3fully	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%
5conv 1fully	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%
5conv 2fully	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%
5conv 3fully	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%
6conv 1fully	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%
6conv 2fully	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%
6conv 3fully	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 2.11: Classification results of modifiednetworks

(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 2.12. 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.

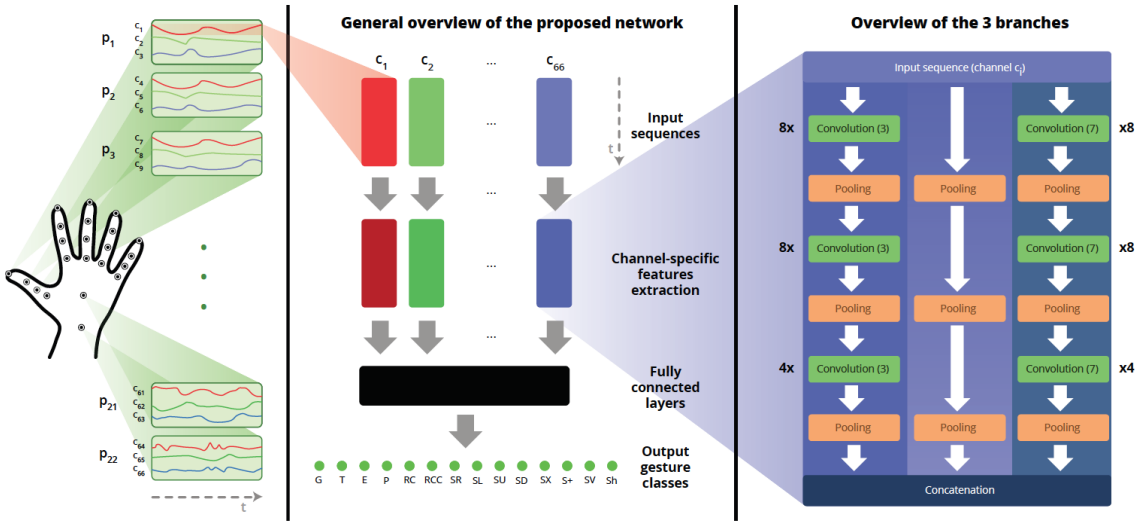


Figure 2.12: parallel convolutional neural network

2.4 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.

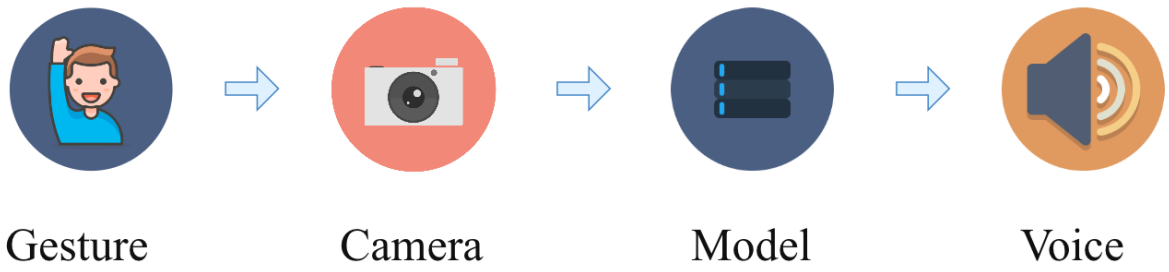


Figure 3.1: System block diagram

3.1 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.

3.1.1 Faster R-CNN

The Faster Region-based Convolutional Network (Faster R-CNN) is a mixture among the Region Proposal Network(RPN)[1] and the Fast R-CNN[2] model.

- A CNN produces feature map form the input images.
- A 3x3 sliding window moves through feature map and and maps it into lower dimension.
- Every sliding window, produces multiple regions based on fixed ration (anchor boxes).
- Each region contain an objectness score and it's bounding box coordinates.

The 2k scores represent the softmax probability of each of the k bounding boxes being on "object." If an anchor box has an "objectness" score above a certain threshold, that box's coordinates (4k coordinates) get passed forward as a region proposal. Then the region proposals are being fed into a Fast R-CNN, followed by a pooling layer, several fully-connected layers and softmax classification layer with bounding box regoessor. Faster R-CNN uses RPN to avoid

[1] algorithm to output bounding boxes to all objects in an image.

[2] A main CNN with multiple convolutional layers is taking the entire image as input instead of using a CNN for each region proposals (R-CNN).

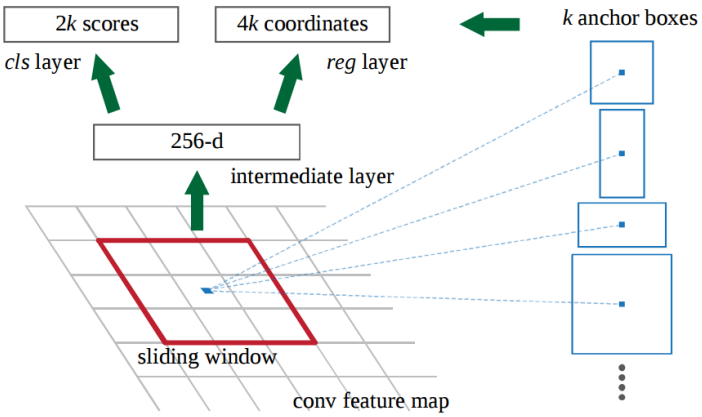


Figure 3.2: One sliding window location. Retrieved from https://towardsdatascience.com

the selective search method [3], it accelerates the training and testing processes, and improve the performances. (Ren, He, Girshick, & Sun, 2017)

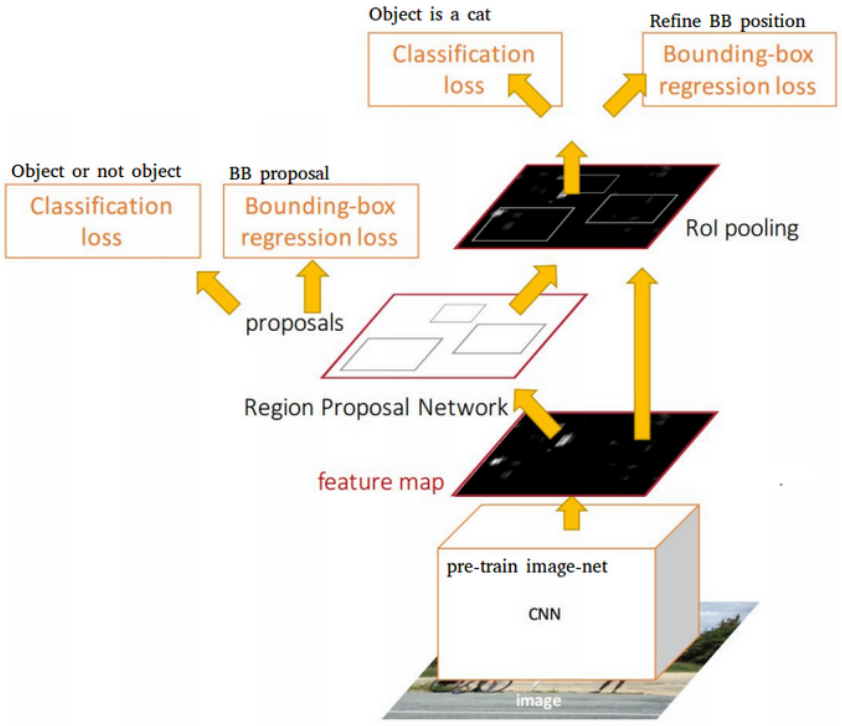


Figure 3.3: Faster R-CNN. Retrieved from https://towardsdatascience.com

[3]Region Proposal algorithm based on grouping of similar region based on color, size, texture and shape compatibility.

3.1.2 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 [4] 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:[5] 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)

[4] Intersection over Union

[5] Object detection methods often output multiple detections which fully or partly cover the same object in an image.

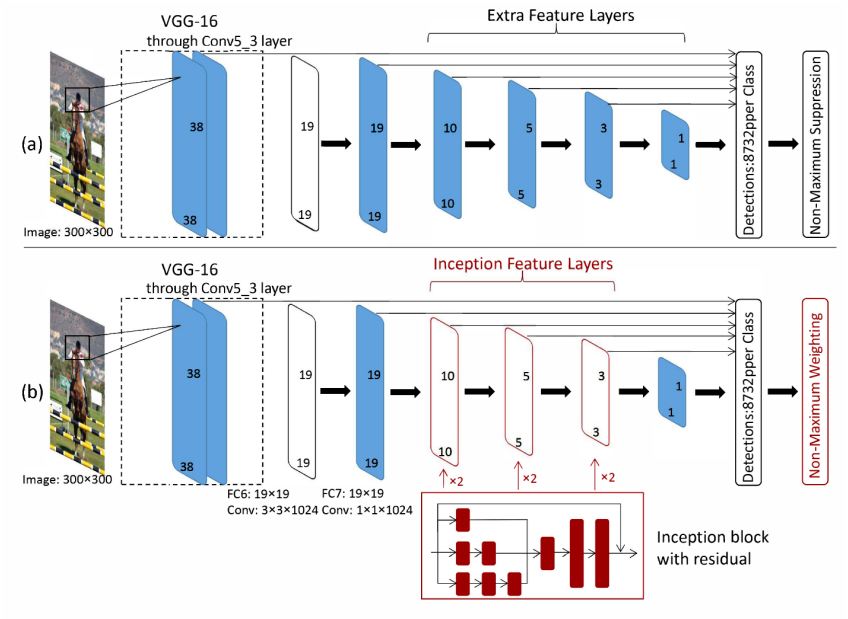


Figure 3.4: SSD. Retrieved from https://www.semanticscholar.org

3.1.3 You Only Look Once (YOLO)

Like SSD, YOLO directly predicts bounding boxes and class probabilities with a single evaluation. The simpleness of YOLO allows real time prediction.

- The model divide the input image into SxS grid.
- Each cell of the grid predict B bounding boxes with a confidence score.
- The score confidence is the probability of detected object multiply by the IoU between the prediction and the truth boxes.

The CNN has 24 convolutional layers followed by 2 connected layers. Reduction layers with 1x1 filters followed by 3x3 convolutional layers replace the initial inception modules.

The Fast YOLO model comes with 9 convolutional layers and less number of filters. The final layer outputs a S*S*(C+B*5) tensor corresponding to the predictions for each cell of the grid. C is the number of estimated probabilities for each class.

Similar to SSD, YOLO predicts so many bounding boxes without any object, So it applies non-maximum suppression method at the end of the network, to merge high overlapping bounding boxes of the same boxes into a single one. The author noticed that still some false positive detected.(Redmon, Divvala, Girshick, & Farhadi, 2016)

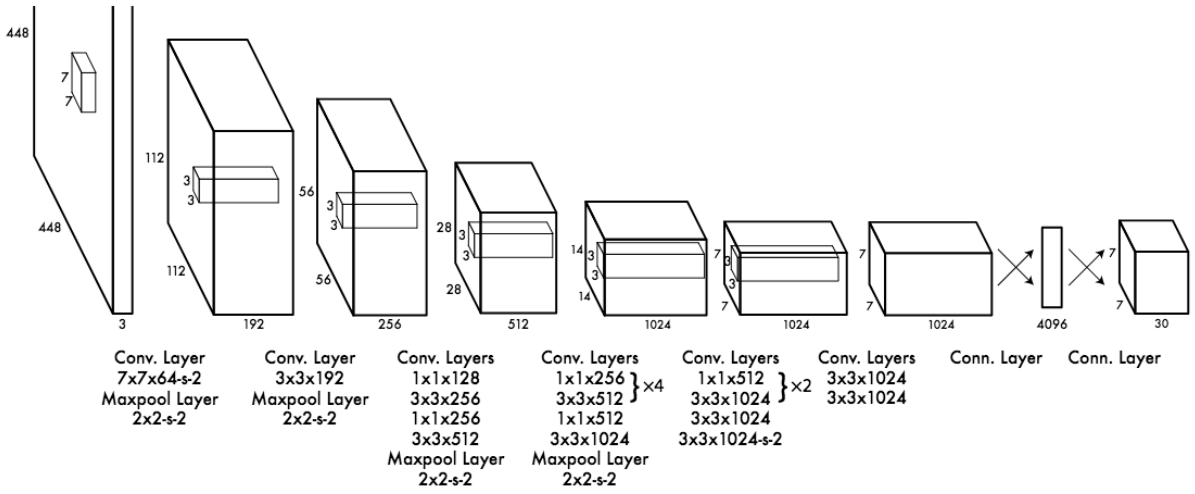


Figure 3.5: YOLO. Retrieved from https://medium.com/

3.2 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.

3.3 Tools

The programming language in use is Python[6] along side with many libraries such as TesorFlow[7], Keras[8], OpenCV[9], NumPy[10], Pandas[11] and Matplotlib[12]. The model is being trained by using Google Cloud Computing[13] service with Ubuntu as operating system.

[6]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/

[7]TensorFlow is an open-source software library for dataflow programming across a range of tasks. https://www.tensorflow.org/

[8]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/

[9]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/

[10]NumPy is the fundamental package for scientific computing with Python. http://www.numpy.org/

[11]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/

[12]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/

[13]Google Compute Engine delivers virtual machines running in Google's innovative data centers and worldwide fiber network. https://cloud.google.com/

References

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

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

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).

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

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, June). You only look once: Unified, real-time object detection. In *Proc. ieee conf. computer vision and pattern recognition (cvpr)* (pp. 779–788). doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017, June). Faster r-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. doi: 10.1109/TPAMI.2016.2577031