Sign language recognition using deep learning

Name: MHD Khaled Maen
Matric No: 1523592

Supervised by
Assoc. Prof. Dr. Amelia Ritahani

October 21, 2018

Contents

Chapter 1

Introduction

1.1 Background

Communication is a process of sending and receiving information among people. Humans communicate with others by a lot of ways but the most effective way is face-to-face communication, Many people believe that the significance of communication is like the importance of breathing. Indeed, communication facilitates the spread of knowledge and forms relationships between people.

Deep learning added a huge boost to the already rapidly developing field of computer vision. With deep learning, a lot of new applications of computer vision techniques have been introduced and are now becoming parts of our everyday lives.

alongside with the power of today's computers, there are now various algorithms that were developed to enable a computers to perform tasks such as object tracking and pattern recognition.

In this study, the focus will be on hand gestures detection and live tracking.

1.2 Problem Statement

The communication between disabled and other people is a really huge gab need to be filled up. In order to overcome this challenge many researches and products have been developed to solve these 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.

Chapter 2

Literature review

2.1 Introduction

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.

2.2 Previous works

Peijun Bao, Ana I. Maqueda, Carlos R. del-Blanco, and Narciso Garcia, (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 2.1. 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.

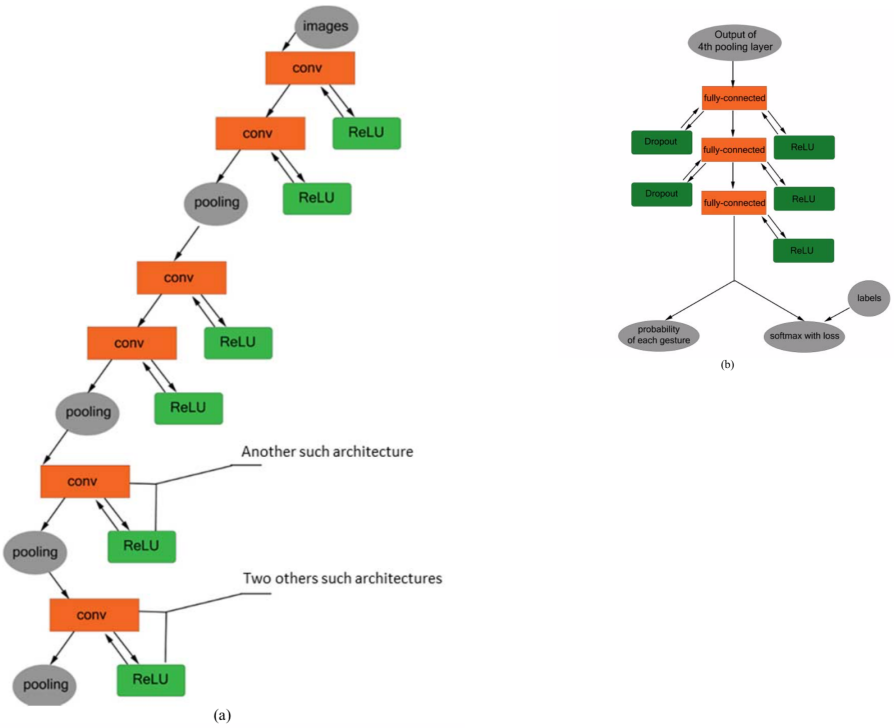

Figure 2.1: Architecture of the proposed deep CNN

G.Anantha Rao, K.Syamala , P.V.V.Kishore, A.S.C.S.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 2.2 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.

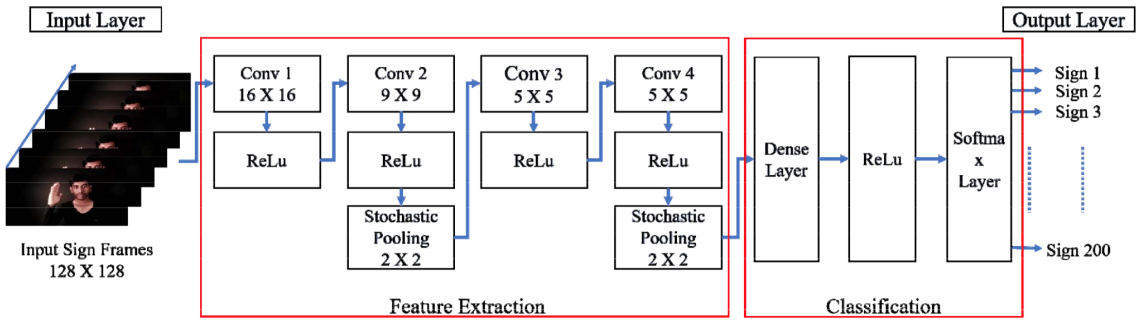

Figure 2.2: Proposed Deep CNN architecture

Soeb Hussain, Rupal Saxena Xie Han, Jameel Ahmed Khan, Prof. Hyunchul Shin, (2018) 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 2.3 as the pretrained model. The According to the paper the accuracy was 93.09%,while using AlexNet figure 2.4 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.

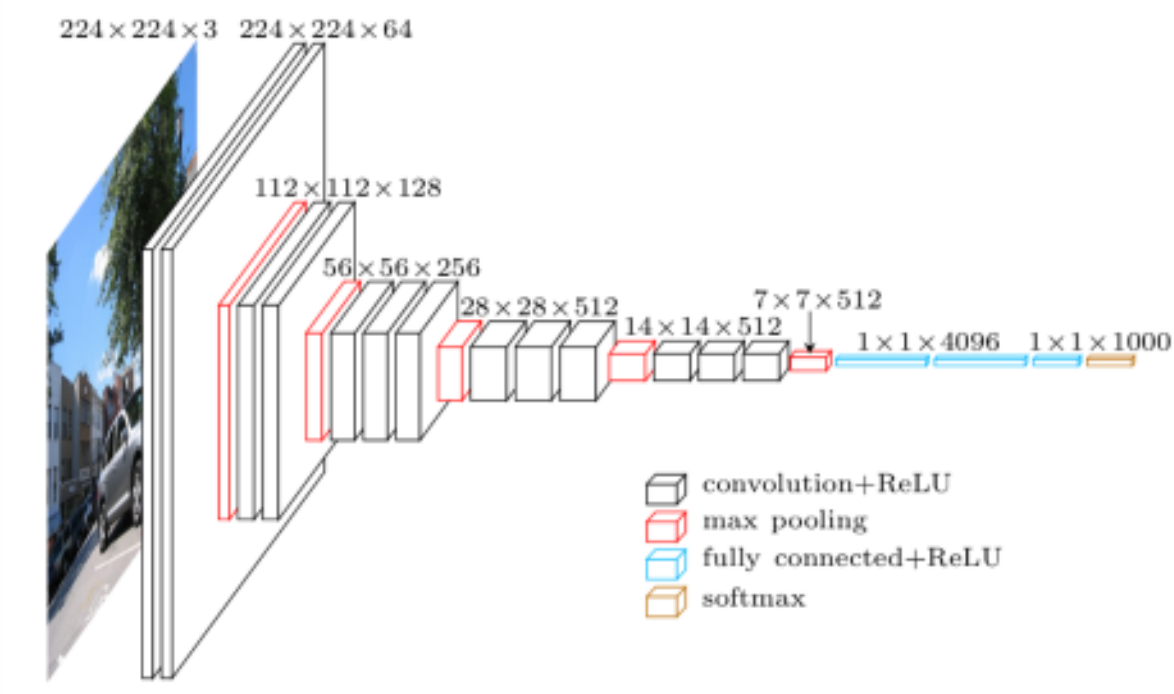

Figure 2.3: VGG16 architecture

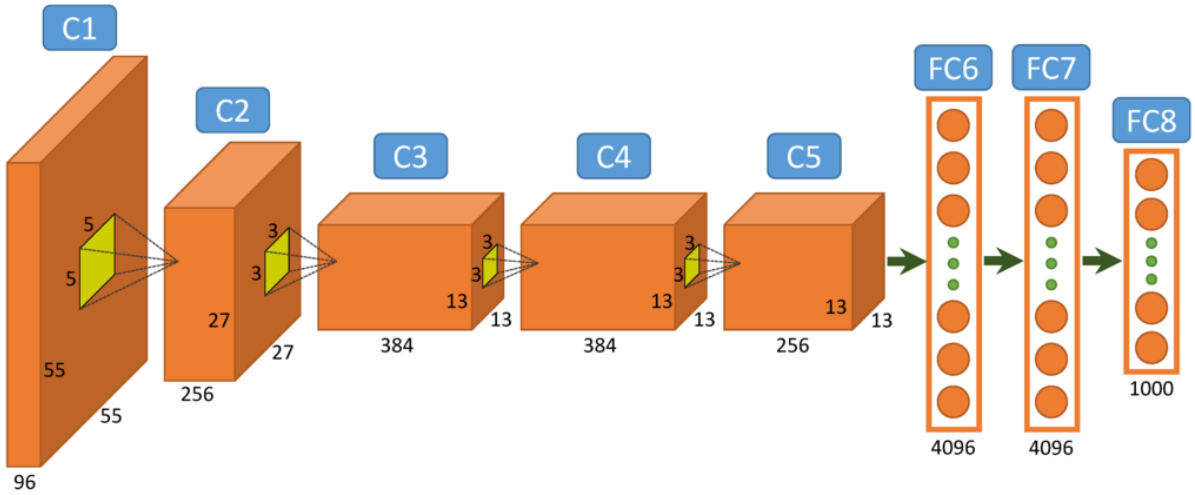

Figure 2.4: VGG16 architecture