Sign Language Recognition using CNN and OpenPose and making 3D Model of Sign Language

by

MD. Jubel Hosssain
16301230
MD. Asaduzzaman Sarker Anik
16301169
Farzana Haque Toma
19241017
Ananya Rahaman
16301159

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

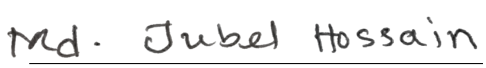
Department of Computer Science and Engineering
Brac University
October 2020

Declaration

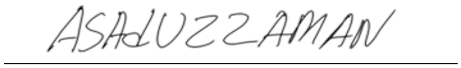
It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

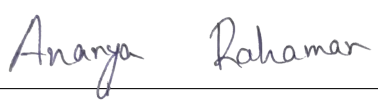


MD. Jubel Hosssain
16301230



MD. Asaduzzaman Sarker Anik
16301169

Farzana Haque Toma
19241017



Ananya Rahaman
16301159

Approval

The thesis/project titled "Sign Language Recognition using Inception v3 and Open Pose and make 3D Model of Sign Language" submitted by

1. MD. Jubel Hosssain (16301230)
2. MD. Asaduzzaman Sarker Anik (16301169)
3. Farzana Haque Toma (19241017)
4. Ananya Rahaman (16301159)

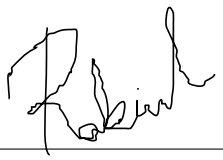
Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 21, 2020.

Examining Committee:

Supervisor:
(Member)

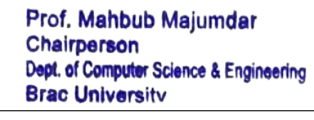
Dr. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)



Dr. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Dr. Mahbubul Alam Majumdar
Professor
Department of Computer Science and Engineering
BRAC University

Abstract

According to the World Health Organization (WHO), 466 million people over the world have impairing hearing misfortune (over 5 percent of the total populace), of whom 34 million are kids. There is an immense correspondence gap between ordinary individuals and hard of hearing and quiet people. Some guardians are unconscious of sending their hearing and talking hindered kid to class or some of them feel bashful. So for learning gesture based communication is hard for those youngsters and some incapacitated grown-up individuals are additionally don't know gesture based communication appropriately. This circumstance causes us to think of our thought. Our venture is making 3D model for all alphabets and numbers as well as some mostly used words and sentences of Bengali language and train the individuals as a virtual teacher. The model will help the individuals until he will do 100 percent correctly. Here for sign language recognition we use algorithm called Inception v3 which is an extended version of CNN (Convolutional Neural Network) and for activity recognition we use OpenPose. Tensorflow is used for coding as this is vastly used for machine learning and deep learning. We use video as our dataset and create 3d avater to teach user sign language. The 3d avatar is created using OpenPose unity plugin. Most of the researches based on sign language are hand gesture based. Unfortunately, sign language not only consist of hand gesture, It also includes face gesture, eye's gesture. And we are taking all of these things in consideration and we are trying to implement all of these. While detecting our Bangla numbers and letters in sign language we have got 89 percent accuracy.

Keywords: Sign language; Machine learning; CNN; Openpose; Tensorflow; WHO; Deep learning; 3D model; Virtual teacher; Unity; Cosine similarity

Dedication (Optional)

We would like to dedicate this thesis to our loving parents and teachers. Also to all who are unable to hear and talk, specially those kids who are physically challenged.

Acknowledgement

Firstly, all praise to the most powerful and most merciful Allah for whom our thesis have been completed without any major interruption. Secondly, to our supervisor Md. Golam Rabiul Alam Sir for his kind support and advice in our work. He helped us whenever we needed help. Even this pandemic, he helped a lot with his precious time. We would also like to thank our family for the mental support that they provided us throughout our whole journey.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Almost 7117 languages are spoken all over the world. There are many books, online platforms and tutors available to teach people those languages. Unfortunately, even in this modern era, people still ignore who are deaf and mute. Many parents feel shy about their children who are physically despair about talking and hearing. They are thought worthless by the society. Parents are inconsiderate about their future and do not give them any chance to enrich their knowledge. Very few parents send their children to special school for learning.
Moreover, in this modern era every learning is online based. So we want to create a platform for those who want to learn sign language online. With the invention of artificial intelligence, people have started desiring to be in the right track more and more. AI proved suitable to solve complex, fuzzy problems starting from playing chess to giving us entertainment to saving our lives by predicting various future events, using neural networks [1].
Unlike spoken language, in sign language people uses hand movement, facial expression, gestures and body language to make other person understand the meaning of what they are trying to say. Mainly people who have impaired hearing, use sign language as a form of communication. Sign language is as enriched as spoken language and this does not depend the spoken language. But sign language has its own sequence of gesture to convey a meaning of something. Sign language is also influenced by the culture and region and just like spoken language, sign language also changes from region to region. So to understand sign language people also need to understand the culture where that person belong. As mentioned before sign language is independent from the spoken language. It has its own grammar, vocabulary and rules to make sentence. Even regions sharing same spoken languages has different sign languages. The diversity it has is the reason working with sign language seems tough. Sign language has both static and dynamic features. Hand shape, movement of hand, orientation of palm and the position of hand are important static feature of sign language. But there are also dynamic features as well, like head position, body posture, and facial expression. We are considering all of these to make a better platform for those who want to learn sign language. This platform will work as a virtual teacher for them to learn sign language.
There are a lots of projects in Kaggle's blog which implemented sign language related research work using deep learning but many of those do not consider head movement,

body posture, and facial expression, eye contact as a whole while recognising sign language. So, we try to implement all of these together and make a platform where people can learn Bangla sign language by a virtual tutor.

1.2 Problem Statement

The purpose of language is to communicate with each other, share ideas and the self-expression of deaf and mute people. There are so many spoken language around the world. Within few kilometers the dialect of the languages varies. Same goes with the sign language as well. So it is important that those who use sign language, they are able to use it correctly. As those who are unable to speak and hear are the ones using sign language, it is very sensitive and important that we communicate with them correctly and they also learn the language accurately.
In sign language, uses of both hands are important. But some words or letters may not necessarily need the use of both hands. Signs that can be described by using one hand is known as static and using both hand is called dynamic. For the right handed people the dominant hand is right and vice versa for the left handed people. We need to consider that even same hand shape or same hand motion can be used to refer different thing base on the movement of the hand. As it is bit tricky to correctly do all those movement and gesture it needs a lot of practice to do it correctly. Even a simple moment can change the meaning completely.
For learning Bangla sign language there are a few institutions and very few school and college. Furthermore, there are not enough materials to learn bangla sign language properly, even on the internet. Which makes communication difficult for the people who uses it. So we want to make a platform which will ease the whole process. We train our model with video input and create a 3D avatar that shows people how to perform Bangla sign language. Then we take video input from user/learner and based on our model show the user how accurately they have performed the sign language. Then judging on their performance they are given feedback to fine tune their ability to perform certain Bangla signs.

1.3 Research Challenges

As most of the researchers worked with image dataset, it is quite difficult for us to work with video dataset. We are required to extract frames from video and from those frames need to generate 3D co-ordinate of users body key-points using openpose unity plugin. And compare these frames with our models video dataset. This part is very challenging as this task need a lot of system memory. So we needed a high performance computer. All of our teammates have laptops and it is really hard to perform processing of a high caliber task like ours on low powered laptops. Furthermore, due to this corona virus pandemic we could not work in our university lab. After accessing remotely in university lab, the network was too slow. So the main hinders of our research was technical issues and resource problem. Moreover, as the video resolution is higher than images, generating x, y co-ordinates of our video was a little bit challenging. Research with Sign language is not a new thing. But research based on Bengali sign language is quite few. We got very few research papers related to Bengali sign language. Datasets from keggale related to Bangla

sign language are non existent or unusable. So we created our own video dataset which were few in numbers and worked with those less amount of data. So, we had to use various data augmentation techniques to increase our own created dataset.

1.4 Research Objectives

Our main goal is to make a virtual system for the people who use Bangla sign language or want learn it. At first the learner has to go to our website. Upon clicking on a word or letter, the system will show the learner a 3D avatar which will perform the sign corresponding to that word or letter. Then the learner will try to copy that letter or word. After that our system will recognize that letter and compare with system's own dataset with cosine similarity. Then tell the user that how accurately or correctly he/she has performed that sign. We want to make a Bangla sign language recognition system using CNN and OpenPose. And also want to make a 3D avatar that teaches Bangla sign language to users. By which we will be able to teach sign language and after learning from the avatar, people will practice by using the pose detection. We will estimate whether the gesture is right or wrong. We use OpenPose for pose detection. For recognizing user's data, we use inception v3 and for making 3D avatar model we used Openpose unity plugin. This will help people to learn sign language accurately and teach them in a proper way. Also if they make any mistake that will be detected too. So the learning and communication will be very much interactive and users will be able to learn in a very practical manner. A learner can learn from home and by himself without any help of a physical teacher or without going to school.

1.5 Organization of Research Paper

From the later section of this research paper The following topics are discussed : Chapter 2 : This chapter discusses previous research involving sign language, OpenPose, CNN, machine learning methods and deep learning schemes. Chapter 3 : This chapter explains the preparation, generation of the indicators, dataset and machine learning used in the model and also explains our model. Chapter 4 : This chapter details the accuracy and performance of the models. Chapter 5 : This chapter concludes the findings and proposes future research.

Chapter 2

Related Work

In this chapter, overview of previous studies of similar works will be discussed in a detailed manner. Mostly about sign language research with openpose, 3D model creating using Unity and research with cosine similarity. Pose estimation is not a new topic for researcher and neither is sign language. It started with Multi-task learning (MTL) was first analyzed in detail by Caruana [2]. Nowadays it is also a great research area. Rajeev et al. worked for face detection, gender detection and landmarks localization. They introduce a method called hyper face using deep convolutional neural networks (CNN). This paper [3] also used multi-tasking algorithm which is Iterative Region Proposals and Landmarks-based NMS. But they use it just for face and gender recognition not for any other implementation where our project can identify all the skeleton joint of a human body. Majumder et al. [4] has researched head pose detection using convolutional neural network (CNN) that they used for making model in low-resolution multi-modal RGB-D data. Sometimes head direction plays a vital role for videos. If we can find the gaze direction then it will be beneficial for us. Moreover, in dark we cannot detect the face of the persons or we cannot detect the activity. But this project [3] uses the combination the two models named classification and regression to estimate regression confidence in a estimated manner. Head pose estimation is also used in two separate and distinct domains, visual surveillance and Human Computer Interactions (HCI) [4]. However, eye direction is also very crucial and we can move eyes in different angles but this paper does not pay any attention to this. This paper is also focused in low resolution scenario not high resolution video whereas our project can perform very well in high resolution video and identify not only head pose but also whole body's pose estimation. Liu et al. [5] worked with bed pose detection. As peoples pose and posture when they are in a bed sleeping or just lying down are important health-related issue and it has a huge potential in many medical applications. There are some diseases which need to monitor when the patients are sleeping such as pressure ulcers, sleep disorder, and carpal tunnel syndrome. Furthermore, in pregnancy period some sleeping pose can be harmful for mother and fetus. As the lighting variation problem is present they use a technique called infrared selective (IRS) image acquisition technique, which was proposed to provide a similar quality data for lighting conditions that varies. They used n-end Histogram of Oriented Gradient(HOG) rectification method so that they can handle the abnormal pose and CNN method. But if the body is covered with blanket or anything else then it could not detect the pose. So this is a major problem of this project. Some

people worked in 2D pose estimation [6] with deep learning and HOG (Histograms of Oriented Gradients). Their aim was to determine the Human pose estimation automatically which can locate the human body parts from images or videos. They worked on heatmap-based framework. Pose estimation opens a new era in medical science and research. Du et al. [7] worked on connecting multi-instrument 2D pose estimation which uses deep neural network. Medical science is still finding it difficult to detect articulation. Nowadays ROBOTIC surgery systems have introduced a power-full platform. With the help if it articulated instruments now can be controlled in a minimally invasive surgery (MIS) through tele-operation of the surgical camera and specialized handheld instruments. This is very important for medical science because in recent years the uses of robot in surgery is increased and in future it has a huge potential. The methodology can be used to detect instruments and by how much degrees they can be moved freely without using any encoder or external sensors. But if they want to track multiple instrument at a time they will fail. On the other hand our project can track multiple object at a time. There is another paper [8] where they wanted to enlarge the range of capture and increase the performance of inter subject and subject to template 3D rigid registration and they use deep learning-based methods. For sign language Najib et al. [9] works with sign language conversion and IoT where they actually did hardware and software combination. They want to train the sign language to deaf and mute people by" Learning Application" and change the sign language into text who want to communicate with them with a special gloves with sensor. In this project they use android application, Hidden Markov Model (HMM) for ML Classification, tensor flow, Arduino Mega for microcontroller, NodeMCU for IoT, Force sensor, Flex sensor, gyro sensor and Rubber woolen gloves. They claim that Previous research has need excess time, memory, money and also inefficient. Despite this, they also faced some kinds of problem. When they use different shape and sizes hand, then error value occurred and also their project cannot detect multiple user at a same time. On the other hand, our project just use software implementation not hardware implementation so we do not have any kinds of problem with the shape of user. Also our project will do the same thing for adult and children without changing any codes or algorithm. Another paper koller et al. [10] implement continuous sign language using Hybrid CNN-HMM. Previously who worked with sign language, they worked with a picture that are not dynamic and they did not count the joint expression of hand and face. If they want use the dynamic environment they need to use hybrid of CNN and HMM because CNN has strong discriminative abilities and HMM has the sequence modelling capabilities. For CNN implementation they use NVIDIA CUDA They use 3 types of data set, these are-1. RWTH-PHOENIX-Weather 2012. But they do not implement their project for multi person users. Their main motive is detect sign language from video when only one person is present in video where our project can detect sign language of multiple person from a video. Farzana [11] et al. worked with Gesture Based Implementation in which Bengali sign language was converted to text with the help of neural network. For this they use Support vector model and for input device they use Microsoft Kinect. They use Visual studio 2015 with a SVM Wrapper for their program implementation. But they worked with isolated sign and static image. This is not applicable for dynamic situation or video and not for multi person uses. Whereas our project can detect Bengali sign language from static and dynamic both situation that means from image and video both and also can detect

multiple user. Actually sign language recognition is not a new research topic but accuracy was not that much good in previous research . For better accuracy Rumi et al.[12] worked with Hand gesture based sign language recognition using inception v3 and vgg16 as image recognition models to train their system Actually inception v3 is a form of CNN(Convolutional neural Network). They take the inputs from live video and images are extracted to be used for recognition. Then their system separates the hand sign from the image and predicts the letters by the model. But their processing is not real time. Because their processing time is little bit high. But our processing time is also faster than their project [11]. Some researcher worked with Automatic recognition of Sign language of Bengali sign language [13]. If any user give any input (sign language) then automatically the project recognize the sign. Chawdhury and kayas implemented their project by using ANN (Artificial neural network), Kinect and OpenNI (open neural Interaction) driver. But they also do not consider dynamic signs or video and multiple user detection from image and video. They also worked with an image to detect an isolated sign. So our project is also better from their project while we consider multiple user and video to accomplish our project. Another project is [14] also worked with continuous sign language detection from video and interpret not only the alphabet but also the whole sentence. But they also do not take consideration of multiple user. If multiple users give input of sign language then the project cannot detect effectively where our project can detect effective while multiple people in a video is present. However, some people worked with multi person activity recognition. Guo and Miao [15] implement this work by using Hidden Markov model (HMM) for classifying multiple persons activities. But their processing time is higher and as they do not use CNN that's why their object detection is little bit harder and they implement their project not for dynamic situation or video whereas we implement our project by detecting multiple person activity recognition in a video. Some researchers [16] also worked with multi person gesture recognition where they built a model to catch these dynamics based on long short-term memory (LSTM) models. Ibrahim et al. represent a deep temporal model with 2-stages for the problem arises from group gesture recognition. Unfortunately, they also do not consider dynamic situation where we consider the dynamic situation as well as multi person activity recognition. In another paper, they[17] look for to set up which calculations and similarity metric combinations can be utilized to upgrade the look and recommendation of articles in a term paper recommender frameworks. Their examination used non-linear classification calculations with content similarity measures. Some paper [18] calculate semantic similarity between documents using cosine similarity.

Chapter 3

Methodology

In this chapter, dataset uses and procedures that are related to this work have been discussed. Overall underlying concept of the algorithms and the related concepts are reviewed and discussed.

3.1 Dataset Description

We have 2 types of dataset in our research.
1.Static dataset
2.Dynamic dataset
For static data we have used images corresponding to bangla alphabets and numbers, and for dynamic data we recorded a bunch of videos of six commonly used bangla words.

3.1.1 Static Dataset

We have used data from previous research named Ishara-lipi dataset on this topic. Fig. 3.1 shows some sample data. The entire dataset is divided into two parts. The first one includes digits and the later one houses characters. The digit dataset has 100 sets of 10 Bangla digit signs and the characters dataset contains 50 sets of 36 Bangla sign character. Bangla Sign Language has 30 consonants and 6 vowels. In this dataset, after preprocessing and discarding errors, 1800 character images and 1000 digit images(total 2800 images) of signs remains and they are in the final state of or dataset.

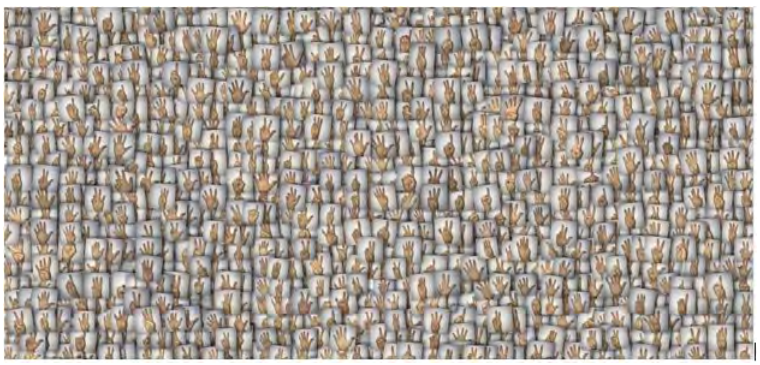


Figure 3.1: Isharalipi Dataset

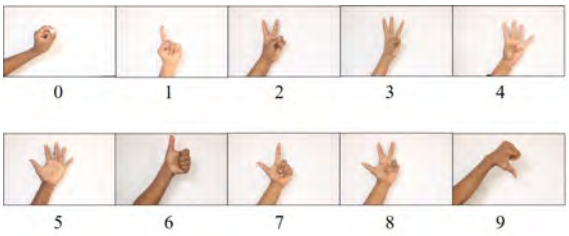


Figure 3.2: Bengali numbers in sign language

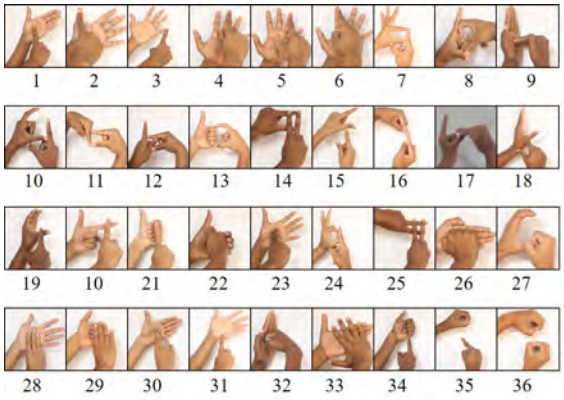


Figure 3.3: Bengali alphabets in sign language

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
অ	আ	ই	উ	এ	ও	ক	খ	গ	ঘ	চ	ছ	জ	ঝ	ট	ঠ	ড	চ	ত	থ

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
দ	ধ	ন	প	ফ	ব	ভ	ম	য়	র	ল	স	হ	ড়	ৎং	ৎঃ

Figure 3.4: Numeric representation of Bangla alphabets

In the figure 3.2 we can see the sign language needed to express the Bangla numbers. There are 10 digits starting from 0 to 9. Then in the figure 3.3 we have shown the sign language needed to show the Bangla letters. We took 6 vowels and 30 consonants. For now, we are just working with the numbers and the letters. The full dataset is divided into two parts- one is for digits(0, 1, 2, 3, . . ., 9) and another is for characters(1, 2, 3, . . ., 36). The dataset that contains digits has ten(0, 1, 2 . . ., 9) Bangla essential sign digits and 100 sets them each, collected from distinctive hard of hearing and common volunteers from diverse establishing. After disposing of most extreme mistakes and performing distinctive preprocessing strategies, 1000 pictures disconnected digits were included within the final dataset. Afterward, a few compelling preprocessing strategies were done so that the information are usable to computer vision and on other application development.

3.1.2 Dynamic Dataset:

We have used dynamic data by using our own video. We have made videos of six words and sentences and used those videos in OpenPose. OpenPose extracts body key-points from those videos and those key-points co-ordinates are used as our dataset. Figure 3.4 shows "Hi".

Figure 3.5: Dynamic sign language

In figure 3.4, we identify the key points of hand and face which is needed to make "Hi".

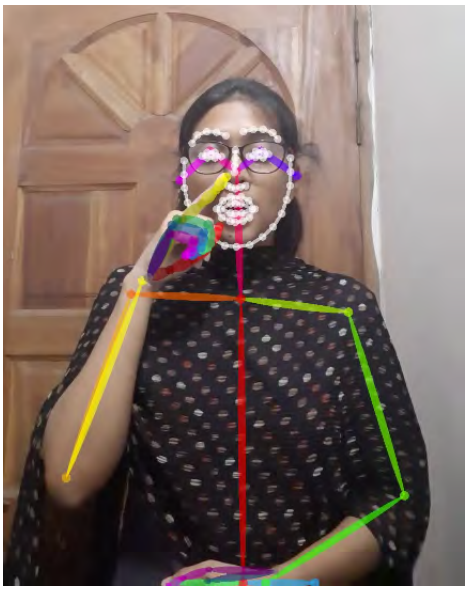


Figure 3.6: Dynamic sign language

In figure 3.5, we identify the key points of hand and face which is needed to make "Mother". Here we also identify the eyes and lips, two hands, nose as this plays an vital role for expressing "Mother".
Below we add the key points of a whole body. Figure 3.6 is taken from OpenPose. Here, all body's key point are shown which starts with from neck and denoted by 1 and ends with ankle and denoted by 24.

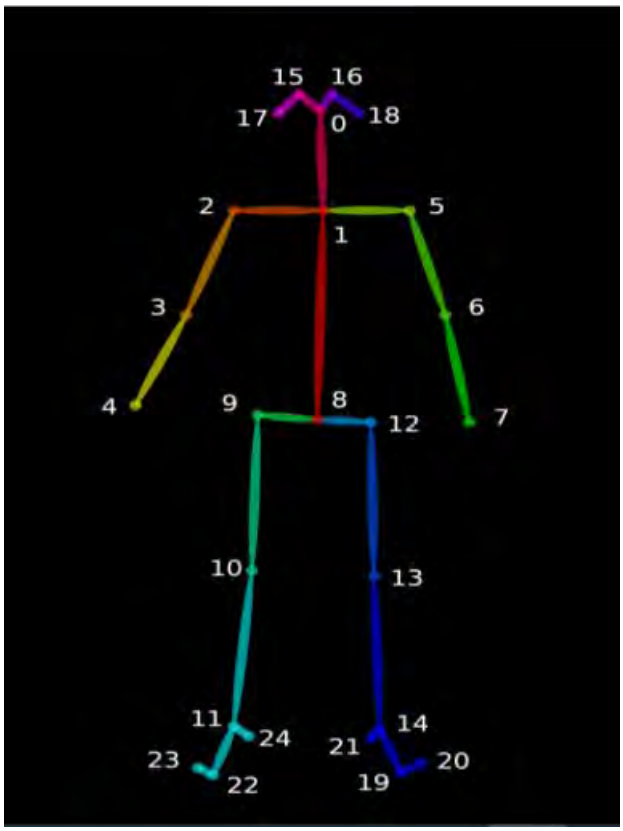


Figure 3.7: Whole body key points

3.2 Workflow

There are two types of BdSL dataset, ones that do not need dynamic hand movements (like the alphabets and numbers), and ones that do (words like ma, baba). Let's call them static and dynamic signs. Actually, in our system, we have done front end design and back-end design for implementing our model. Moreover, for back-end design, we worked with static data and dynamic data.

3.2.1 Front-end design

At first, user have to go to our website to learn the sign language. In our website, we have various options like letter, digit, word, sentence etc. Then he should select from that option what he wants to learn. For our model to interact with the user we have created a website. It is a simple website that enables users to learn bangla sign language interactively. The homepage of the website consists of options showing users what they can learn or the available words or signs. The user can select or search for individual words, alphabets or numbers or they can select an option that enables them to learn from the very basics and guide them to more complex signs. When a user selects a word he/she will be redirected to another webpage that will play a video of that sign. The user can replay the video as many times as he/she wants. When he/she is ready to practice, he/she will press the practice button. This will again take him to another webpage. This page will have 2 video windows. One will play the sign video, and another will show the user through their webcam. The user will press the try button and a countdown timer of three seconds will be

shown to prepare the user. After the three seconds timer, the user will proceed to do the sign in front of his/her webcam and by using cosine similarity a result will be shown indicating how well or accurately the user has performed the sign. If the cosine similarity is close to 1 then the webpage will prompt the user that he/she has done the sign perfectly and he/she can try learning a new sign or he/she can again practice the same sign. If the cosine similarity is not satisfactory, the user can retry or skip to learn another sign.

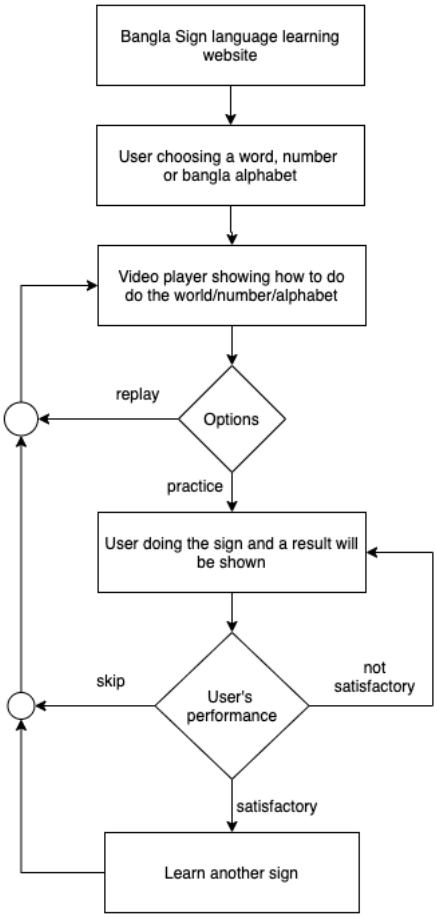


Figure 3.8: Flow chart of front end design

3.2.2 Back-end design

We use two types of back-end design. One is for static dataset and another is for dynamic dataset.

Static sign language: Since static signs do not involve dynamic movements, it can be classified with images. We will need a dataset with images of those static signs and using them we can create our model to classify those signs.

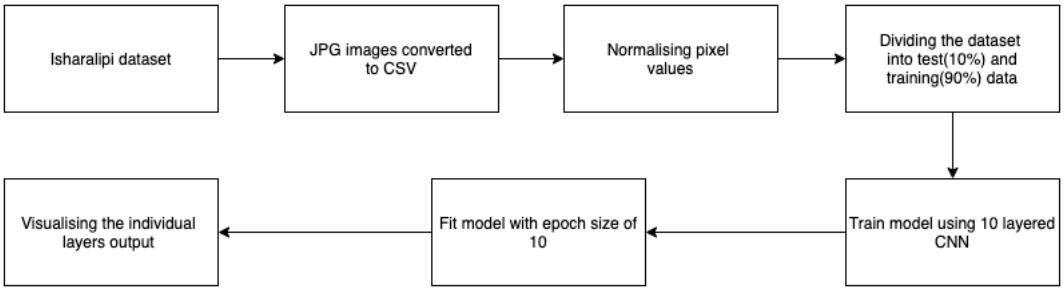


Figure 3.9: Workflow of static sign language

The isharalipi dataset includes images of bangla vowels, consonants and numbers. They are in JPG format. For our use we converted it to CSV format. The images are sized at 128*128 pixel. So converting them to CSV file create rows with 16384 columns. Then we divided the dataset into two parts, 10% for training data and 90% for test data.. The labels for those images are assigned by numeric values and they indicate individual vowels, consonentents or words associated with our dataset. Then we normalise the pixel values and proceed to create our model.
We added ten layers for our model. The model is sequential. The first layer is Conv2D with filter value of 32, kernel size of (5,5) and we have used rectified linear unit or relu activation function. The second layer is MaxPoll2D with pool size of (2,2). Then the third layer is dropout to reduce overfitting. The fourth layer is again Conv2D but this time the filter number is increased to 64, kernel size is decreased to (3,3) and the activation function is set to relu. After that another maxPool2D layer is added with pool size of (2,2) and strides of (2,2) on the fifth layer. The sixth layer is dropout. The seventh layer is flatten. Then a dense layer is added with activation function relu. The second last layer is dropout and the final layer is dense with activation function softmax. After adding the layers we fine tune it with some post processing. Then we set the epoch value to 10, batch size to 64 and summarise the model. Finally we fit the model to our dataset and evaluate it. We create a confusion matrix to visualise the prediction output for all the alphabets.

Dynamic sign language:
For dynamic signs we have created our own dataset. We have recorded multiple videos for each of our six selected signs as our dynamic dataset. We convert those videos into sequences of JPEG images and pass them through openpose. Openpose returns keypoint values from each image frame in JSON format. We then store these JSON values into a numpy array of shape (300, 150, 390). Here 300 indicates the total number of videos in our dataset, 150 is the maximum number of frames per video and 390 is the amount of data points returned by openpose for a single image. Since we have converted video to images we need to label images properly so that our model knows what to predict. To do that we look through the sequence of images and mark the ones that corresponds to the end of that sign. We then created a numpy array of zeros with the shape of (300, 150, 6), here 6 is one hot vector corresponding to the alphabetical ordering of the words of it's sign. We have 300 video data and we split it 80% and 20% so that 240 videos are for training and the remaining 60 are for testing our model. Then we feed 14040000 (240*150*390) inputs to the first layer of our RNN model. First we feed the data to two LSTM

layers with 128 units each. After that, we do the same with another LSTM layer with 64 units. Finally we put everything through a time distributing dense layer with an activation function of softmax, which results in 6 element vector.

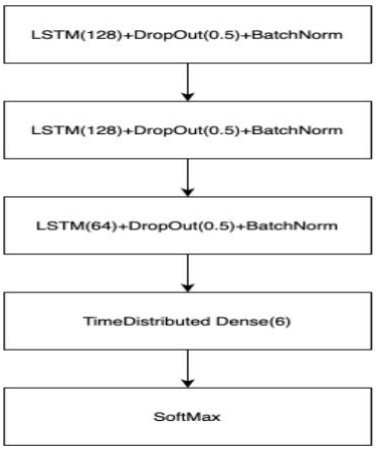


Figure 3.10: RNN Architecture

We choose the softmax activation function in the last layer since we want our model to output probability, not strict prediction. This also helps us to figure out where our model is making mistakes and its confidence in its prediction. The model has 50 percent dropouts after the four LSTM layers. Then we use Batch normalization after the dropout. We look for the sign ending image so the model can stop taking input images after the sign is done. The model uses categorical cross-entropy as the loss function. Our RNN architecture has a 6 dimensional softmax output and our ground truth 6 dimensional vector is one hot, thus this loss function of our model works great.

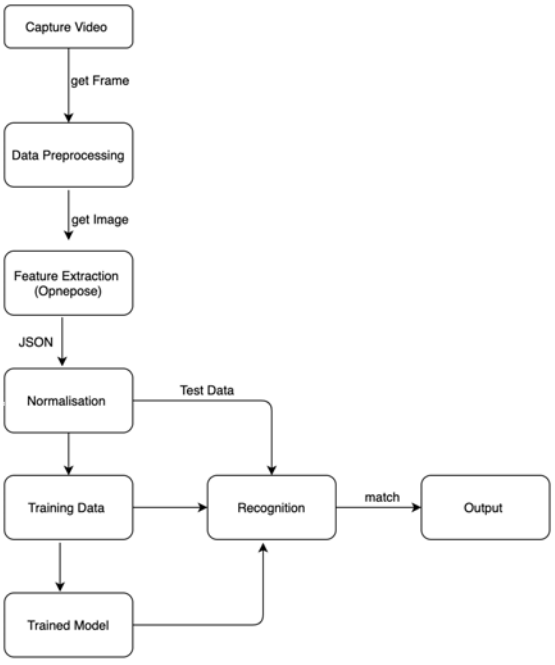


Figure 3.11: Workflow of dynamic sign language

3.3 Artificial Neural Network

Artificial Neural Systems (ANN) are one of the well-established tools of machine learning ponder. The word "neural" recommend the neurons of human brain. This component is propelled from the brain of a human. This method mirrors the operation of a human brain. The neurons of brain can alter their behavior based on chemical response or the electrical signal which they are given. The way of learning through neural network is dynamic. Comparable to human learning system, this strategy contains the input. It can be greatly complicated. Connections of each network have distinctive parameters. By calibrating the weight of the systems as well as changing the parameters of the systems can give diverse state. In the figure 4.1speaks to a neural network. The left layer of this neural network is known as input layer. The circles speak to the input to the organize. The center layer is known as hidden layer. The reason of hidden layer is, the esteem of it cannot be found within the training set. Furthest right layer is the yield layer. In this case, output layer has only unit. There can be different unit in the output layer.

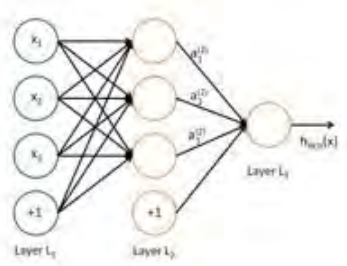


Figure 3.12: Artificial Neural Network

$$a_1^{(2)} = f\left(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}\right) \tag{3.1}$$

$$a_2^{(2)} = f\left(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}\right) \tag{3.2}$$

$$a_3^{(2)} = f\left(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}\right) \tag{3.3}$$

$$h_{W,b}(x) = a_1^{(3)} = f\left(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)}\right) \tag{3.4}$$

Some circles are labeled here as "+1". This units are called bias units. Bias units have only outgoing connection to other layers. However, there are no incoming connection to a bias unit. So, the value of it is not influenced by the values of any previous layers. The output without bias unit can be written as function output = w * input (y = mx).

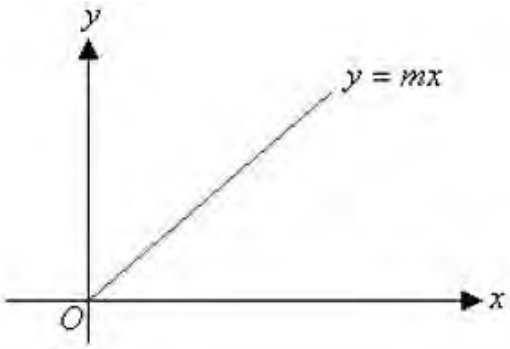


Figure 3.13: Straight Line

When we change the weight of the function, the gradient of the function changes to make it steeper. As a result, it can affect crucially to a number of problems. An optimal model may not pass through the origin. To setting y-intercept, the function needs to be changed accordingly. The function needs a bias unit so that it can have constant it the function. Now the function is output = w * input +wb(y = mx+c). Here, w= weight and b= bias. The wb denotes the term c.

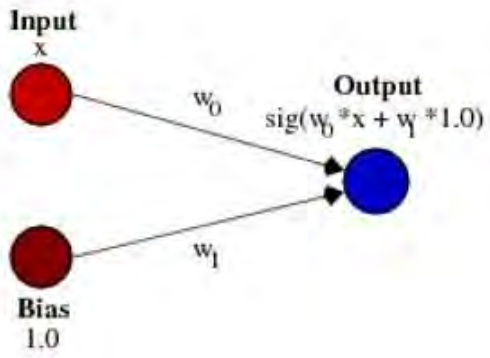


Figure 3.14: Bias

From the above diagram, we can see that in weighted sum we added the bias term and weight wb , and then fed into activation function. The constant term is known as “intercept term”. This moves the activation function either left or to right. If the input is zero then this is the output. Figure 3.16 shows how different weights will transform the activation.

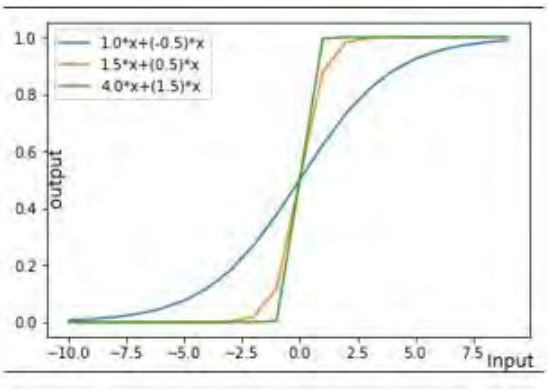


Figure 3.15: Graph of activation function triggering without bias

If we add the bias unit, then the possibility of activation function will be shown like below picture:

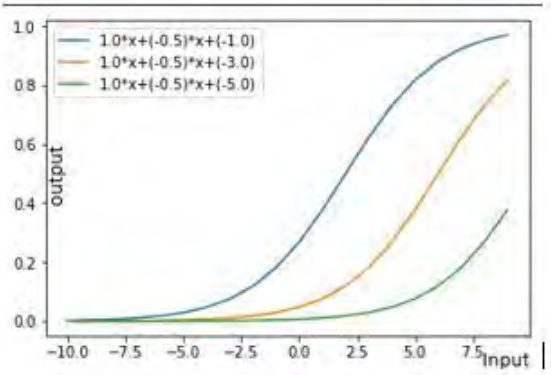


Figure 3.16: Graph of activation function triggering without bias

3.4 Backpropagation

Backpropagation is a common method by which a neural network is trained. This simplifies the network structure through weighted links which has less impact on the network. It uses the utility of chain and power rules. The focus of this method is to minimize the cost function by regulating the biases and weights.

Model Initialization Neural networks can begin from anywhere, as in genetic algorithms and theory of evolution. Thus, a model's random initialization is a common practice. The reasoning behind this is that it can reach to the pseudo-ideal model from wherever it begins, if it is sufficiently perseverant and through an iterative learning process.

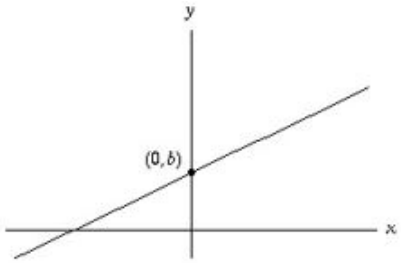


Figure 3.17: linear regression intercept

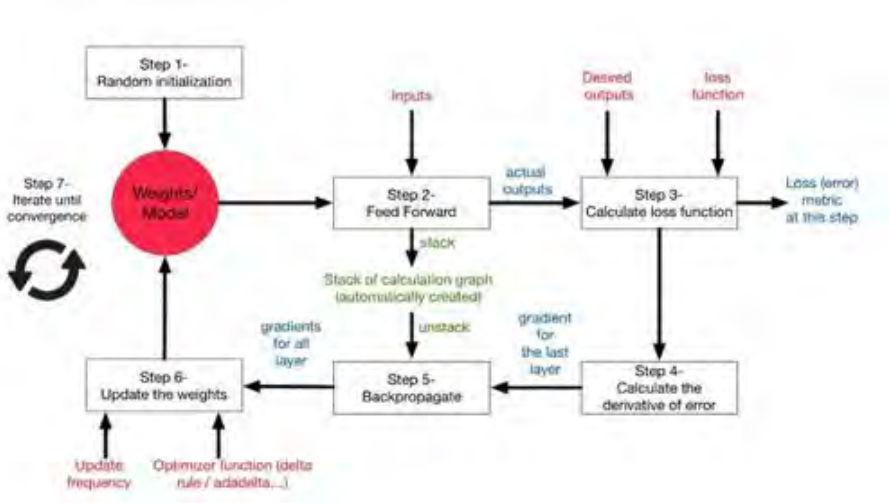


Figure 3.18: Overview of neural networks learning process

Forward propagation After initializing the model at random, the next natural step to take is to test its efficiency. Taking the input of the network and pass through it to the network layer and measure the model's actual output directly. This phase is called forward propagation as the calculation stream goes in the direction forward from the input to neural network then to the output [19]. Loss function If a problem is possible to generalize that is what called loss function. If a problem is possible to generalize that is what called loss function. It is essentially a performance metric on how well the NN performs to achieve its goal of generating results as close to the target values as possible [19]. The straightforward approach of the function equation is

$$function = desiredoutput - actualoutput \tag{3.5}$$

However, the function provides positive values when prediction less than desired output. If the opposite is true i.e. if the prediction greater than desired output, the function provides negative value. To reflect the absolute error, it can be often written as

$$function = Absolute(desiredoutput - actualoutput) \tag{3.6}$$

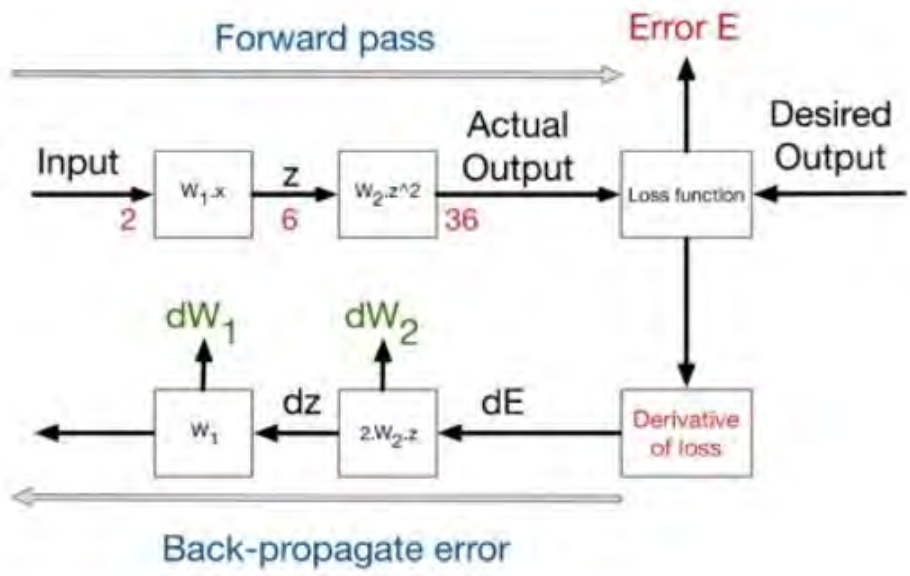


Figure 3.19: Diagram of forward and backward paths

Under some circumstances, the total error of the function can be same for summation of a number of small errors or summation of few big errors. As it is a prediction work, it is preferable to have the distribution of a number of small errors than a few errors of larger values. Here lamba/2m is regularization term. h-theta is output term.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[y_k^{(i)}\log\left(\left(h_\theta\left(x^{(i)}\right)\right)_k\right) + \left(1-y_k^{(i)}\right)\log\left(1-\left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{n}\sum_{j=1}^{n_{li}}\left(\theta_{ji}^{(l)}\right)^2 \quad (3.7)$$

Gradient Descent: This is a method that helps to know the w(weight) and b(bias) parameters in a way that minimizes cost of the function. Generally, the cost function of logistic regression is convex in nature. It has only one global minima. This is the reason for taking this function. The squared error can have more than one minima.

3.5 Deep Neural Network

Deep learning is part of an artificial neural network-based component of machine learning methods. Deep-learning networks are differentiated by their level of detail

```
do

   for Each training example named example

      prediction = neural-net-output (network, example) //
      forward pass

      actual = teacher-output(example)

      compute error (prediction - actual) at the output units

      compute {displaystyle Delta w_{h}} for all weights from
      hidden layer to output layer // backward pass

      compute {displaystyle Delta w_{i}} for all weights from
      input layer to hidden layer   // backward pass continued

      update network weights // input layer not modified by error
     estimate

until every example classified correctly or any stopping
criterion satisfied

return the network
```

Figure 3.20: Pseudo Algorithm

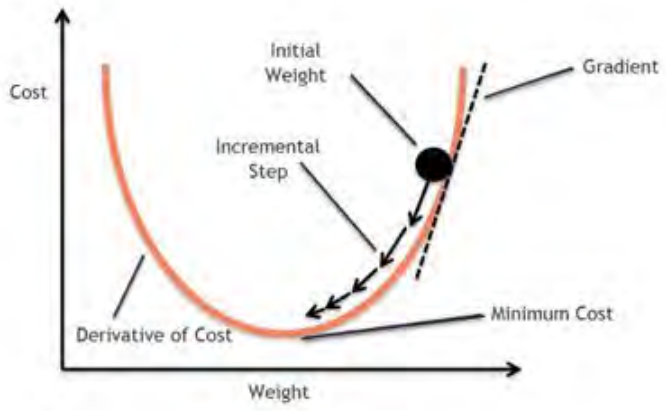


Figure 3.21: Gradient Descent

from the more general single-layer neural networks. Older neural network models such as the first representations were shallow consisting of one input and one output layer, and at most one hidden layer in between. Over three stages of learning (including input and output) count as" deep" learning. Deep neural network can overshadow the performance of shallow feed neural network. The reason is, in a shallow network, it has lower number of hidden layers. It produces less accurate results than the deep neural network.

3.6 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a very tools of machine learning and Artificial intelligence. Images recognition, images classifications can be done by CNN. We can also perform object detection, face recognition by using it. We have to use input data (images) and that data will be passed through some layers which is called convolution layers and convolute those with karnels. Then we use max pooling or

average pooling and use those for fully connected layer. At the last step we use softmax for probabilistic value and that value vary from 0 to 1. We can understand the whole CNN process from the below figure

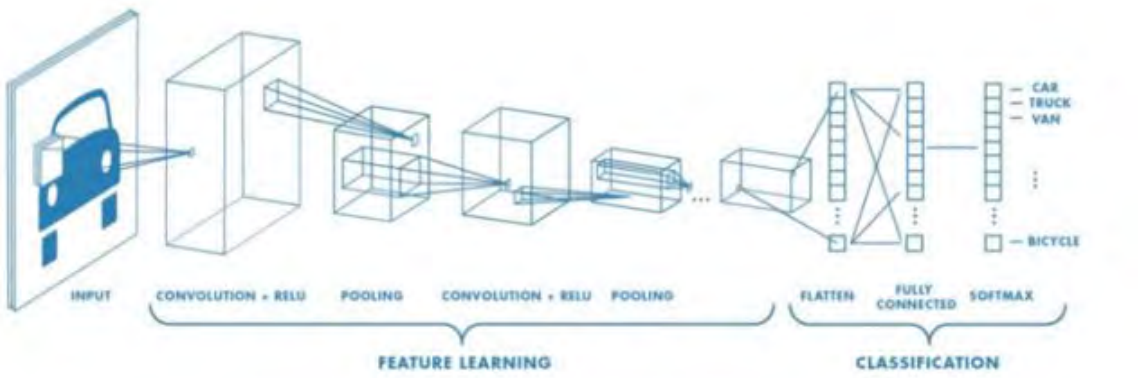


Figure 3.22

The first layer of CNN is Convolution It extract the features of image from input. Then the whole image divided into small pixels which is squared shape. The small square learn the image feature by using convolution. Convolution is a image processing tool where it takes inputs as image matrix and use filter.

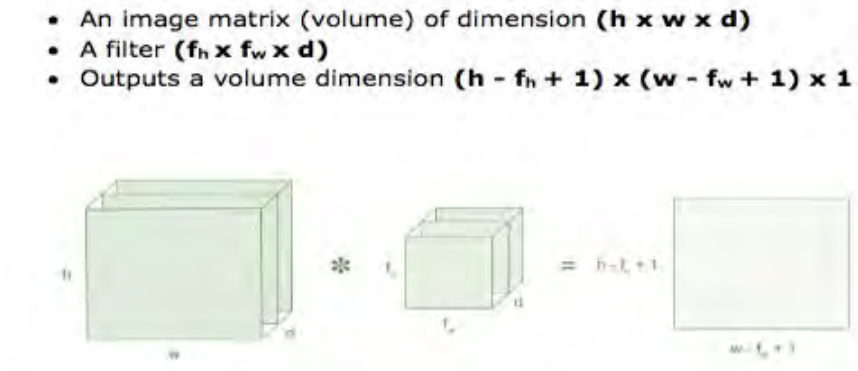


Figure 3.23: Image matrix multiplies kernel or filter matrix

If you take a 5 x 5 matrix of a image and it has only 0, 1 values and consider a filter matrix 3 x 3 as shown in below

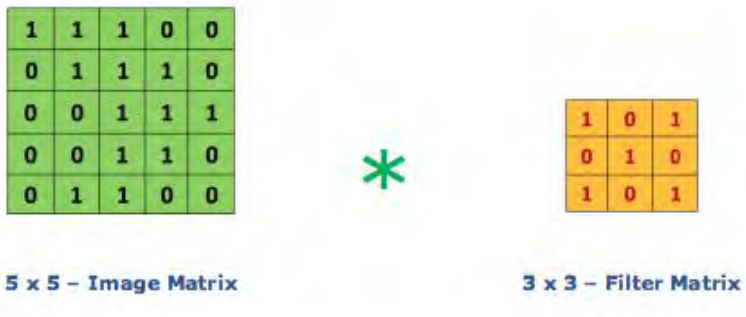


Figure 3.24: Image convoluted by filter matrix

Then we can multiplies 5 x 5 image matrix with 3 x 3 filter matrix and this process is called as "Feature Map". The below picture shows this process.

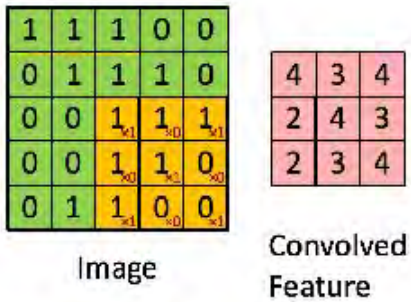


Figure 3.25: Convoluted feature

By using convolution process with different filters, we can detect edge, we can make blur of an image and sharpen an image. With the below example, we can visualize different convoluted image after applying various filters.

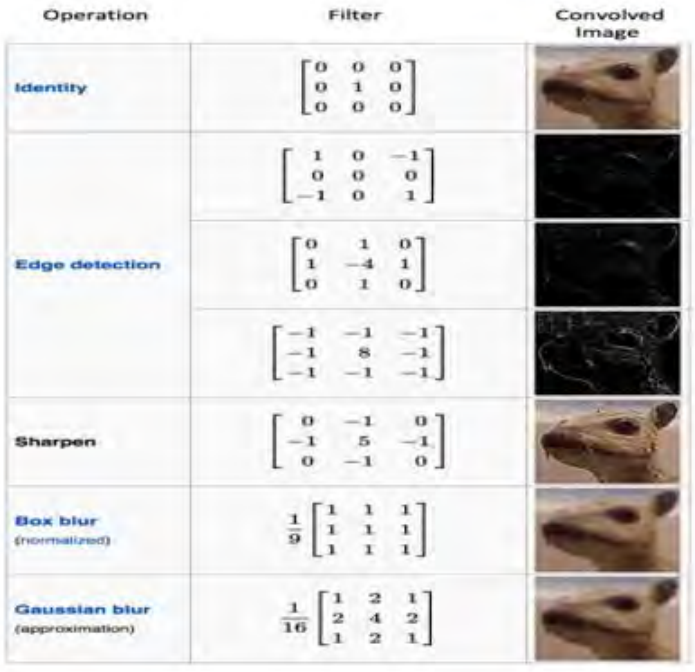


Figure 3.26: Some popular filter matrix

The number of pixels in a input matrix which is moved in a input is called stride. If we make our stride 1 then it shifts the filter 1 pixel. If we make our stride 2 then it shifts the filter 2 times. If we make our stride 3 then it shifts the filter 3 times and so on. The below image illustrate the whole process by using 2 stride.

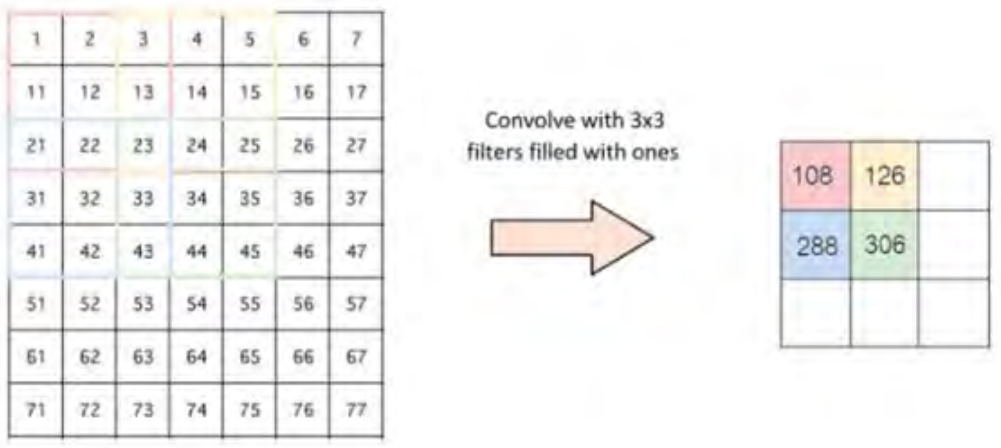


Figure 3.27

After doing the above process, we apply padding on the input image. But sometimes the filters we apply in the image does not fit properly. Then we can do tow thing. One is using zero padding and another one is discard the portion of the image where the image part is not fit properly. After using padding we use ReLU. The full form of ReLU is Rectified Linear Unit for a non-linear operation. The equation of ReLU is f(x) = max(0,x). ReLU is very essential for CNN because it familiarizes non linearity of our convolution network. Because we always expect positive linear value for our image

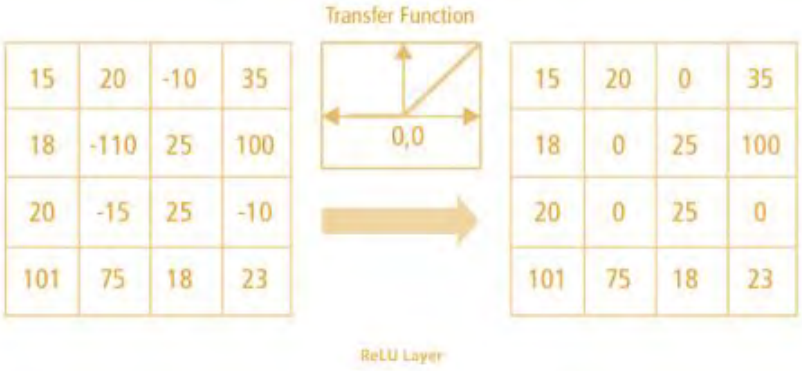


Figure 3.28: ReLU operation

We can also use some other non linear application like sigmoid, tanh etc. We can also use those function. But maximum research use ReLU because the performance of ReLU is much better than sigmoid and tanh.

After ReLU operation we use Pooling layer. As we have many pixels in a image and that's why we have many features in the network. Sometimes we need to discard some features to make easy our computation. If the main features are needed to reduce or the dimensions ae needed to reduce then we perform pooling. Here we hold the main features. this is called as subsampling . we use many kinds of pooling

types such as Max pooling, average pooling, sum pooling Taking the largest elements from the feature map is known as Max pooling. If we use the average of all elements of the feature map is called as average pooling. sum pooling is taking the sum of all the elements of feature matrix.

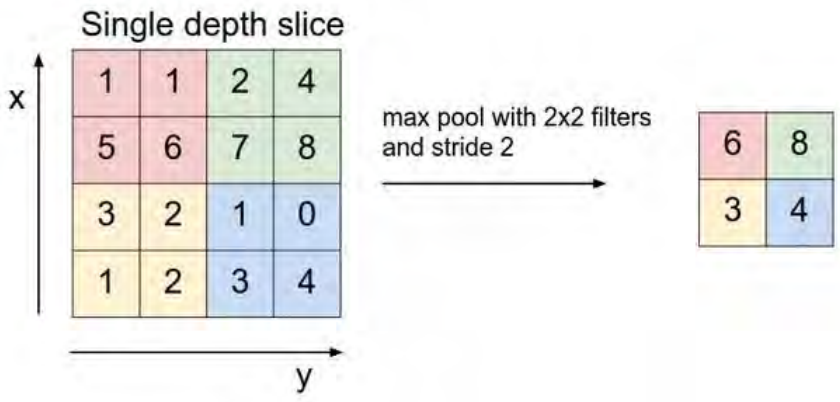


Figure 3.29: Max Pooling

After doing the pooling performance then we need to perform FC layer processing. The full form of FC layer is Fully connected layer. We need to flatten our matrix into vectors and use that as normal neural network.

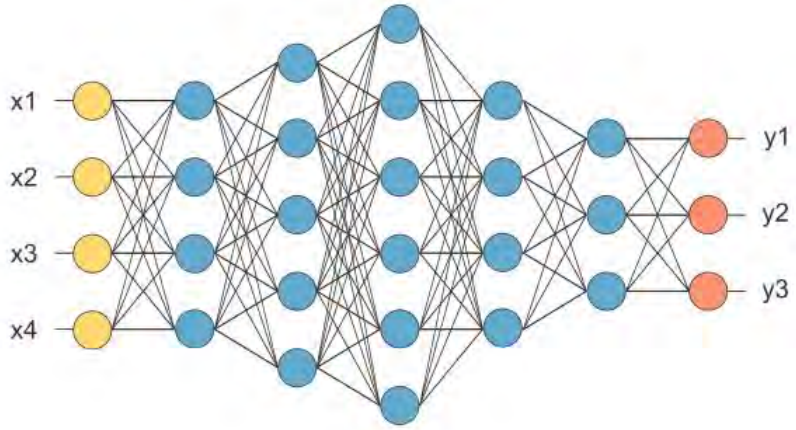


Figure 3.30: After pooling layer, flattened as FC layer

From the above figure, we converted the feature map into a vector. For creating a model we need to attached all the features together from FC layer. At last for classifying our model, we need to use a activation function for example softmax or sigmoid to predict something.

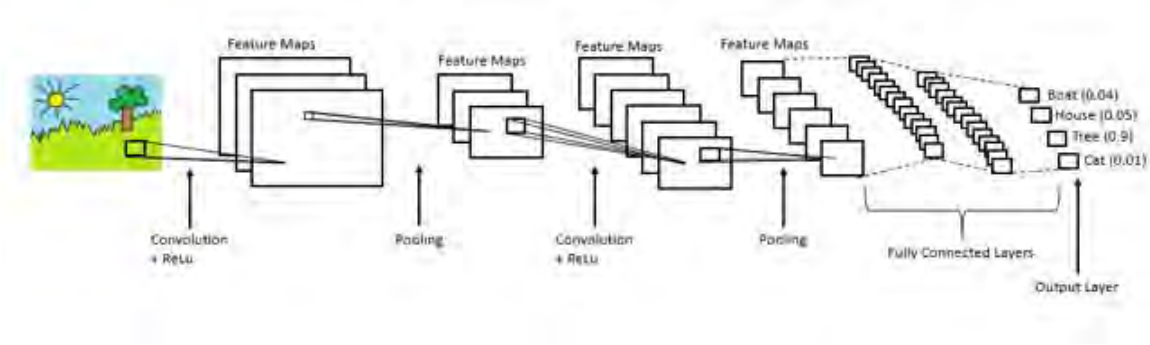


Figure 3.31: Architecture of CNN

Now we can summarize CNN :

1. Take input dataset to perform CNN.

2. then we need to choose parameters, apply filters with strides, apply different padding. Perform convolution on the image and apply ReLU activation to the matrix.

3. Use pooling layer (Max pooling/Average pooling/sum pooling).

4. Use Convolution layer until not satisfying the condition.

5. Flatten the output which was got from pooling layer and use fully connected layer (FC Layer).

6. Predict the object from image by using softmax or sigmoid function.

3.7 Inception v3

Inception v3 is done by using CNN. If we want to make our model more efficient in terms of computation and need to reduce more parameter then we need to use inception v3. Actually, it uses transfer learning for completing its task.

Transfer Learning

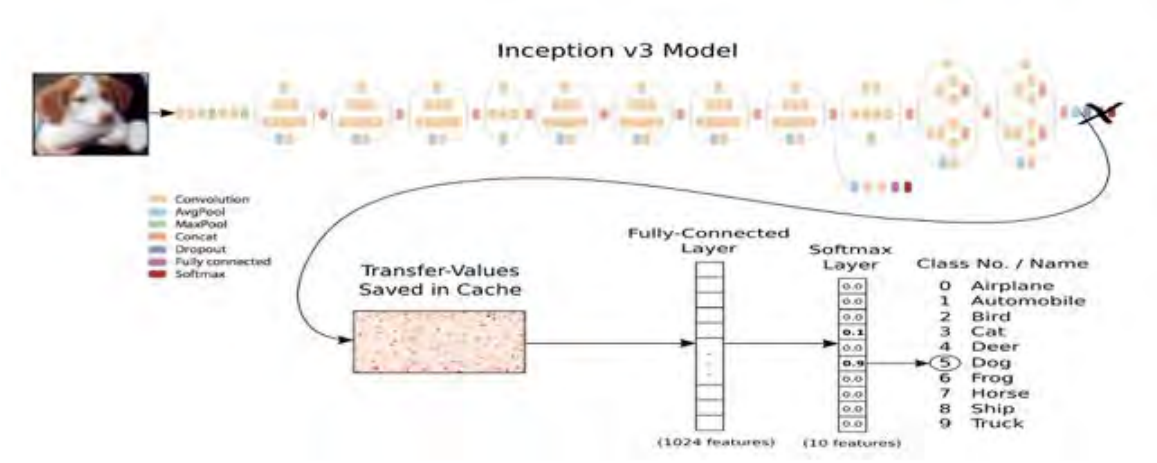


Figure 3.32: full workflow of Inception V3 and Transfer Learning (Source)

Transfer learning is a process which allows to train the final layer again in a network and it uses the previous knowledge from the network. That decreases the time to train model and reduces the datasets. In inception v3, transfer learning is used for training the model. from the above mentioned, inception v3 is used for training some large powerful dataset where millions of data present and need to reduce computational cost and reduce time to train data. This is a very useful where we want to use large dataset and we need to perform CNN. In inception v3 , you can able to maintain the previous layer knowledge that the model learnt through training and can apply in a smaller dataset

3.8 OpenPose

OpenPose [20] is a multi-person pose detection system that can distinguish and detect human facial, body, hand and feet key-points(total 135 key-points) in real-time on a single image or video. It includes API and plugins for Python, C++ execution and Unity.

Architecture:

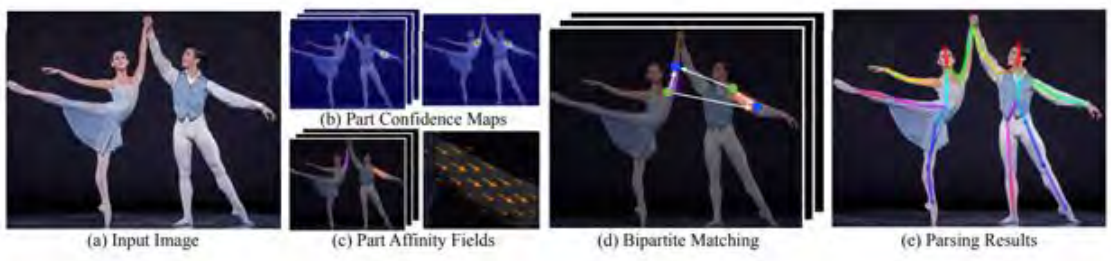


Figure 3.33: Picture using open pose

Figure 3.34 shows the way OpenpPose extracts key-points from a single image.

1. In the initial step an input image is being passed through a baseline CNN network that extract the provided input image's feature maps. The author has used a VGG-19 network that houses 10 layers.

2. After getting the feature map, it is processed in a CNN pipeline with multiple stages. This generated Part Confidence Maps and Part Affinity Field
   • Part Confidence Maps • Part Affinity Field

3. Finally, A greedy bipartite matching algorithm is used on the previously generated Part Confidence Maps and Part Affinity Fields to get the poses for every person on the input image.

Confidence Maps and Part Affinity Fields: Confidence Maps: A 2D representation of whether a specific body part can be located in any of the given pixel of the input image is called a Confidence Map. Figure 3.4 describes Confidence Maps.

$$S = (S_1, S_2 \ldots, S_J) \text{ where } S_j \epsilon R^{w*h}, je1_{\ldots J} \tag{3.8}$$

Here J is the representative number of body parts locations.

Part Affinity Fields: The Part Affinity is a combination set of 2D vector fields. It looks for and detect different persons limbs location and orientation in the input image. It then encodes the data in a pairwise connection for the same person body parts.

Multi Stage CNN:
The multi-CNN architecture described previously has three major steps:

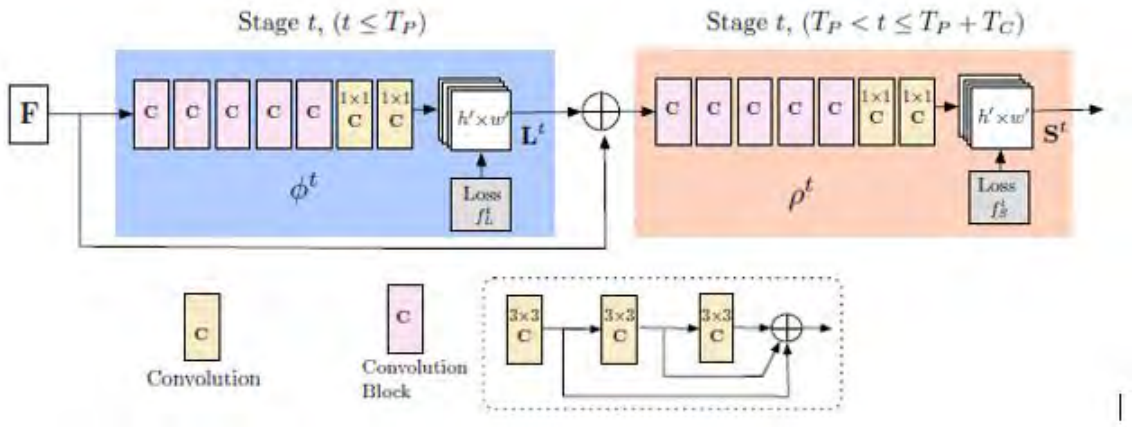


Figure 3.34: Background in OpenPose

• In the beginning set of stages, it predicts the Part Affinity Fields that refines Lt from the feature maps of the base network F.

$$\mathbf{L}^t = \phi^t \left(\mathbf{F}, \mathbf{L}^{t-1}\right), \forall 2 \leq t \leq T_P \tag{3.9}$$

• The later set of stages use the output from the previous layers Part Affinity Fields to polish prediction of the detection of confidence maps.

$$\mathbf{S}^{T_P} = \rho^t \left(\mathbf{F}, \mathbf{L}^{T_P}\right), \forall t = T_P, \tag{3.10}$$

$$\mathbf{S}^t = \rho^t \left(\mathbf{F}, \mathbf{L}^{T_P}, \mathbf{S}^{t-1}\right), \forall T_P < t \leq T_P + T_C \tag{3.11}$$

• In the final stage, S (confidence maps) and L (Part Affinity Field) go through the greedy algorithm for further process.

Loss functions:
There is an L2-loss function that is used to evaluate the loss between the predicted confidence maps and Part Affinity fields to the ground truth maps and fields.

$$f_{\mathrm{L}}^{t_i} = \sum_{c=1}^{C} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \left\| \mathbf{L}_c^{t_i}(\mathbf{p}) - \mathbf{L}_c^*(\mathbf{p}) \right\|_2^2 \tag{3.12}$$

$$f_{\mathrm{S}}^{t_k} = \sum_{j=1}^{J} \sum_{\mathrm{p}} \mathbf{W}(\mathbf{p}) \cdot \left\| \mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p}) \right\|_2^2 \tag{3.13}$$

In figure 3.8, Lc* represents the ground truth part affinity fields, Sj* signifies the ground truth part confidence map, and W is a binary mask with W(p) = 0 when the annotation is missing at the pixel p. This is done so that the extra loss generated by these mask can be prevented.

There are intermediate supervision at every stage that is used to address the problem arising from vanishing gradient by reloading the gradient periodically.

$$f = \sum_{t=1}^{T_P} f_L^t + \sum_{t=T_P+1}^{T_P+T_C} f_S^t \tag{3.14}$$

Confidence Maps: The Confidence maps for every person k and every body part j can be represented with figure 3.11.

$$\mathbf{S}_{j,k}^*(\mathbf{p}) = \exp\left(-\frac{\|\mathbf{p} - \mathbf{x}_{j,k}\|_2^2}{\sigma^2}\right) \tag{3.15}$$

Figure 3.11 is a Gaussian curve that has gradual changes where sigma controls the spreading of the peak. The predicted peak of the network is an aggregation of the individual confidence maps by a max operator.

Part Affinity Fields:

The part affinity field is required especially in multi person pose detection we are required to map the correct body parts to its body. Because for multiple persons, there are multiple heads, hands, shoulders etc. Thus it becomes difficult to distinguish sometimes when they closely grouped together. PAF provides a connection between different part of the body that belongs to the same person. A stronger PAF link between body parts represents that high chances that those body parts belong to the same person.

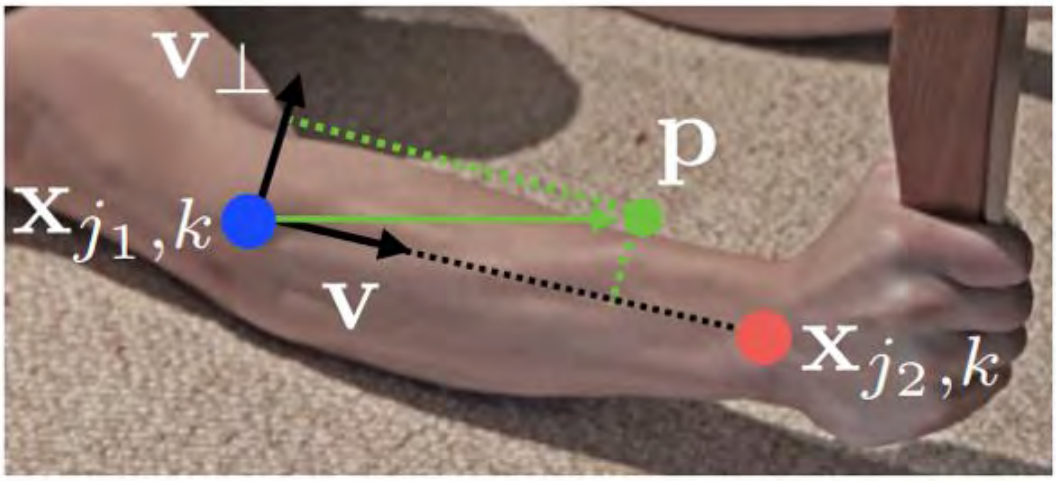


Figure 3.35: x,y,z co-ordinate of a key points of a hand

• If the p is on the limb, then L* is the unit vector otherwise it is 0.

$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c,k \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

$$\mathbf{v} = (\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}) \,/\, \|\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}\|_2 \tag{3.17}$$

• The predicted Part affinity field, Lc along the line segment is to measure the confidence for two candidate part locations dj1 and dj2:

$$E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{\|\mathbf{d}_{j_2} - \mathbf{d}_{j_1}\|_2} du \tag{3.18}$$

• For multi person, Total E needs to be maximized :

$$\max_{\mathcal{Z}_c} E_c = \max_{\mathcal{Z}_c} \sum_{m \in \mathcal{D}_{j_1}} \sum_{n \in \mathcal{D}_{j_2}} E_{mn} \cdot z^{mn}_{j_1 j_2} \tag{3.19}$$

• There are multiple approaches to connect the body part as shown in the image below:

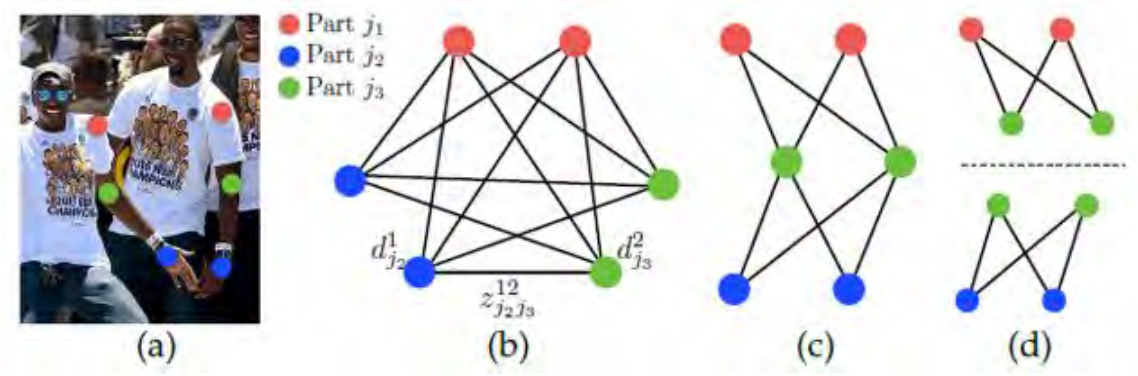


Figure 3.36: Hands key points

- Association by Body part detection.
- Association by considering all edges and generate a k-partite graph
- Association by generating the tree structure.
- Association by generating different bi-partite graph using greedy algorithm.

Changes from CMUPose:
CMUPose is the earlier version of OpenPose. It is the architecture that won the COCO 2016 Key point detection challenge 2016.

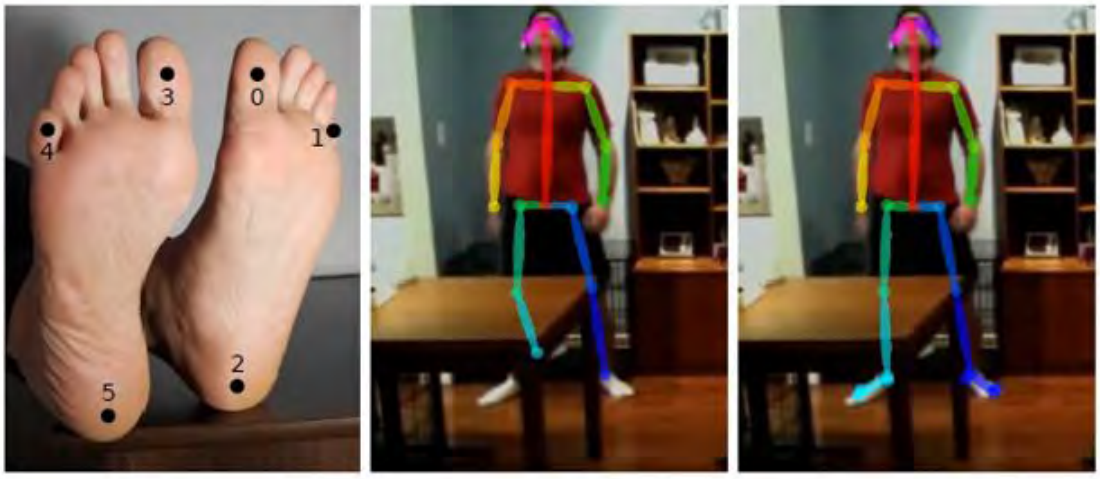


Figure 3.37: Human detection in OpenPose

OpenPose also proposed a foot detection algorithm. It makes OpenPose the first combined body and foot keypoint dataset and detector. By adding that it is able to detect ankle more accurately.

Figure 3.38: Car detection in OpenPose

Similar to body Pose detection, the author of OpenPose experimented this algorithm on Vehicle Detection. It records high Average Precision and Recall on that.

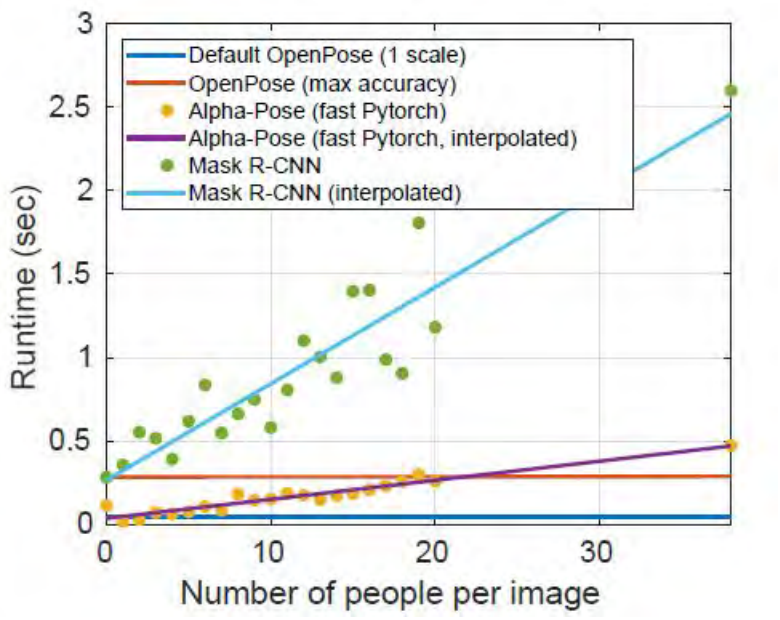


Figure 3.39: Graph of number of people

Fig. 15: Common failure cases: (a) rare pose or appearance, (b) missing or false parts detection, (c) overlapping parts, i.e., part detections shared by two persons, (d) wrong connection associating parts from two persons, (e-f): false positives on statues or animals.

Fig. 16: Common foot failure cases: (a) foot or leg occluded by the body, (b) foot or leg occluded by another object, (c) foot visible but leg occluded, (d) shoe and foot not aligned, (e): false negatives when foot visible but rest of the body occluded, (f): soles of their feet are usually not detected (rare in training), (g): swap between right and left body parts.

Figure 3.40: Human in different positions

• OpenPose have problems estimating pose when the ground truth example has non typical poses and upside down examples.
• In highly crowded images where people are overlapping, the approach tends to merge annotations from different people, while missing others, due to the overlapping PAFs that make the greedy multi-person parsing fail

3.9 Unity

Unity is the world's driving stage software for making and working intelligently, real-time 3D substance, giving the apparatuses to form astounding games and publish them to a wide extend of gadgets. The Unity center stage empowers whole inventive

groups to be more profitable together. Unity gives users the capacity to make games and encounters in both 2D and 3D, and the motor offers an essential scripting API in C, for both the Unity editor within the frame of plugins, and games themselves, as well as drag and drop functionality. Earlier to C being the essential programming dialect utilized for the motor, it already bolstered Boo, which was expelled with the discharge of Unity 5. We use unity for making 3D vision.

3.10 Cosign similarity:

If we want to measure the similarity between two different text, audio, object etc. then we use cosine similarity. It measures the cosine angle of a certain point between two positive vectors of a certain vector space. This is only care about orientation of two object rather magnitude. If the two vectors are in the same introduction then the cosine similarity will be 1 and if the vectors are in opposite introduction then the cosine similarity is zero. On the other hand, if the two vectors are inversely situated then the similarity will be -1 as they are oppositely oriented. This process actually measures between 0 and 1. Actually generally cosine similarity is used for positive space .Cosine similarity is not depends on value(Magnitude), degree. It only depends on orientation.

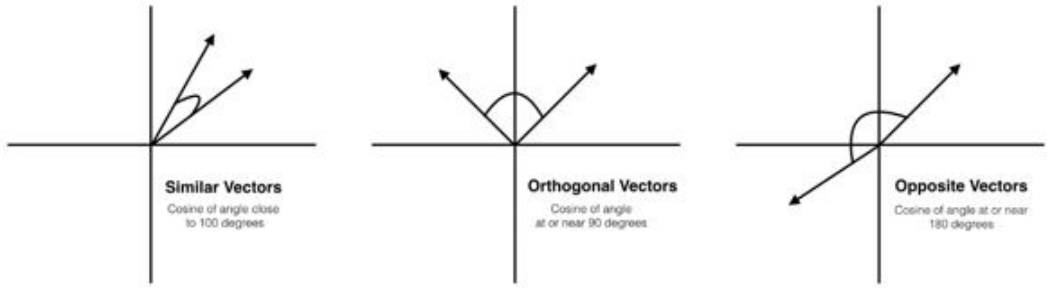


Figure 3.41: Cosign similarity

How does Cosine Similarity Work?
The Cosine Similarity measures the orientation of two different vector. We can find the dot product of vector mathematics and the formula can be written as:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\|\|\mathbf{B}\| \cos\theta \tag{3.20}$$

We can also define cosine similarity as follows:

$$\text{similarity} \ = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.21}$$

The output will be in range of -1 to 1, where -1 means totally opposite or non similar, 0 is perpendicular, and 1 means 100 percent similarity.

Cosine similarity application For expanding past abstract mathematics, Cosine Similarity has used for very long time . Information retrieving, data mining, data

recovery, and matching the two content is very popular uses of cosine similarity. If the answer is greater than zero and less or equal to 1 then we can say that the contents are similar. The higher the value , the higher similarity. Here we can detect how similar two texts are and this similarity does not depend on size. In a multi-dimensional space , it estimate the cosine angle of two vector. The advantage of cosine similarity is even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. If the angle is small then the similarity will be higher.

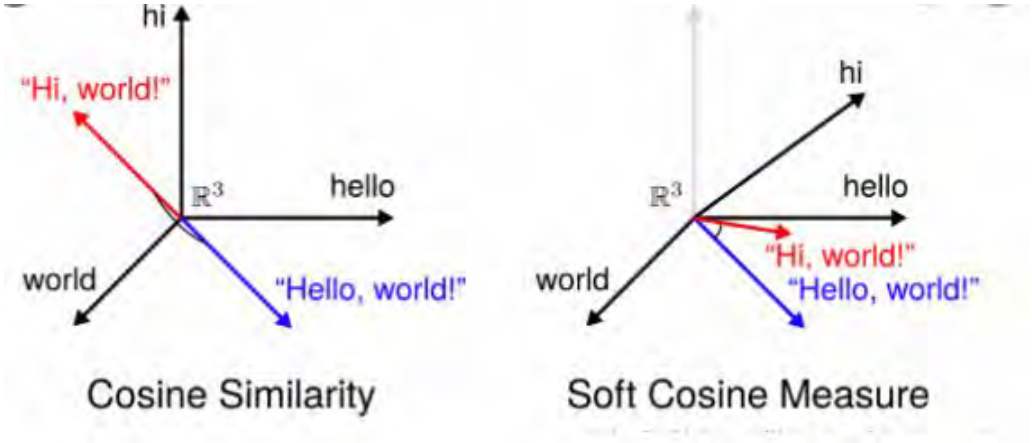


Figure 3.42: Cosine similarity of 2 words

3.11 Recurrent Neural Networks

When a network recall the past values and according to their previous information their action are driven and informed then that network is called Recurrent Neural Network. Basic feed networks still remember things, but they remember what they discovered during learning process. A regular neural network is used as an input in a fixed size vector which restricts its use in situations involving an input with no predetermined length.

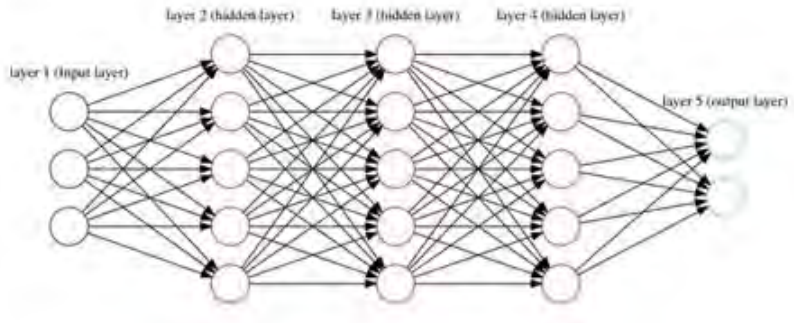


Figure 3.43: Deep layered (3) architecture

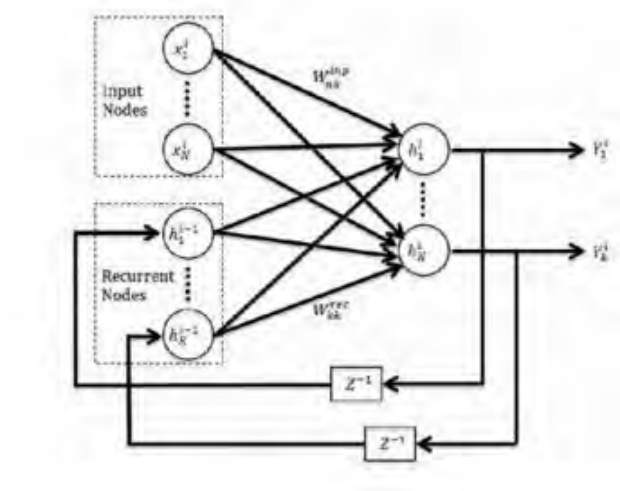


Figure 3.44: Recurrent Neural Network

The whole process depends on time particularly. the output of the hidden layer is reused in RNN. It fed back the output to itself. Loss function — In RNN, the loss function L is calculated by using the loss at every time step. This function is for minimizing the cost of the model.

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}\left(\hat{y}^{<t>}, y^{<t>}\right) \tag{3.22}$$

Backpropagation using time- we can perform Backpropagation in time for each point. If we consider timestep T, the derivative of the loss L, weight matrix W , then the equation is-

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \left. \frac{\partial \mathcal{L}^{(T)}}{\partial W} \right|_{(t)} \tag{3.23}$$

The problem with basic recurrent neural networks

Regular recurrent neural networks are not used very often [20]. The key rationale is the issue of the vanishing gradient. Ideally, it is preferable to have long memories for recurrent neural networks so that the network can link information relationships in time at significant distances. With more time steps, chances of backpropagation going out of the context. The gradient, with more steps, going towards zero. As a consequence, the significance of this value is not impactful. The weight adjustment is also less accurate. So, it does not learn from previous data with more time. Thus, regular recurrent neural network are not commonly used. With long short term memory cell, this problem can overcome.

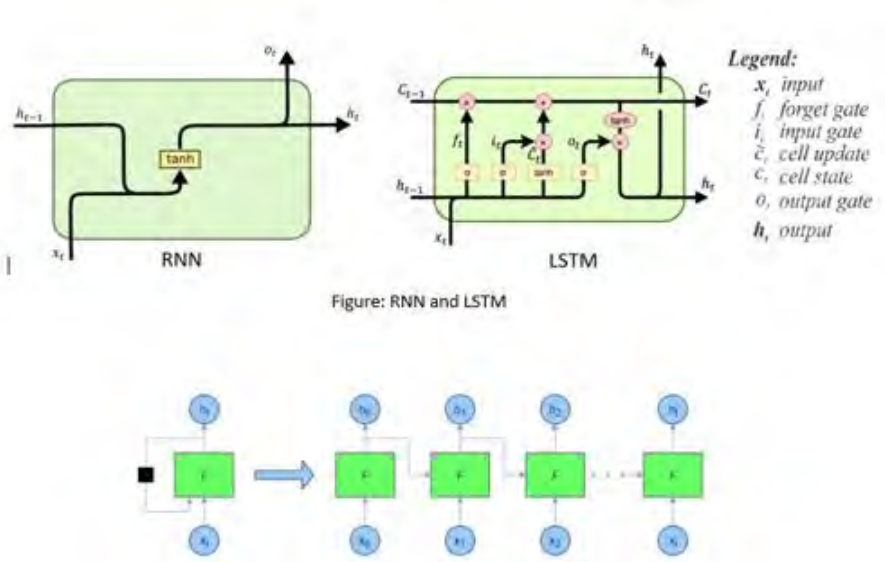


Figure: RNN and LSTM

Figure 3.45: Unrolled Recurrent Neural Network

3.12 LSTM

Long Short Term Memory (LSTM) networks are a adaptation of Recurrent Neural Arrange (RNN). This strategy is able of learning long term encounter from its expanded memory. This systems are well fitted, where the crevice between the data is long, within the issue. From their past setting, LSTM can interface with later information. It can take a single information point and prepare it to arrangements of information through its input association. Long Brief Term Memory fundamentally contains three sorts of entryways which are input door layer, disregard door layer, yield door layer.

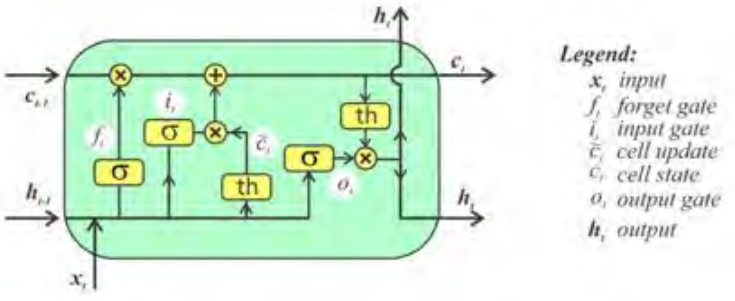


Figure 3.46: General Representation of LSTM

In each neural network cell, the computation has 2 steps.

$$z = WTx + B \tag{3.24}$$

$$propageted_output, a = activation_function(z) \tag{3.25}$$

Here, W is the coefficient matrix. It is shared across all the layers. The intercept matrix is represented with b. T represents the layer. X denotes the input, or, the

previous layer output. There is activation function. it can be sigmoid, tanh, ReLU, or leaky ReLU. A forget gate layer, to decide to discard information, implements a sigmoid function layer.

$$f_t = \sigma\left(Wf\left[h_{t-1}, X_t\right] + bf\right) \tag{3.26}$$

Then, an input gate layer decides which values are needed to be updated. This is also a sigmoid function.

$$it = \sigma\left(Wih_{t-1}, X_t\right] + b_i) \tag{3.27}$$

After that, new candidate values, Ct are created as a vector by a tanh layer. These values are added to the state.

$$C_t = \tanh\left(W_\theta\left[h_{t-1}, X_t\right] + b_c\right) \tag{3.28}$$

Then the final Ct calculated discarding the ft and propagated to a sigmoid function and a tanh function to get our final output.

$$C_t = f_t \times C_{r-1} + it * C_t \tag{3.29}$$

$$ot = \sigma\left(W_a\left[h_{t-1,X_t}\right] + b_{\mathcal{Q}}\right) \tag{3.30}$$

$$h_t = \emptyset_t * \tanh\left(C_t\right) \tag{3.31}$$

Chapter 4

Experimental Results

4.1 Performance Evaluation

Performance analysis metrics help to assess how the established model works with unknown data. It also helps to set confidence bound in future data predictions. In this domain, for anticipating task we want our models to predict each quarter's acquisition premium with high accuracy and precision. For regression analysis, analysts are concerned with how far or close the predicted y is from the actual y. In the graph, if X-axis has the real y value and Y-axis has predicted y, for best case y=yˆ. To measure how good or bad the model predicts, there are several metrics used in statistics.

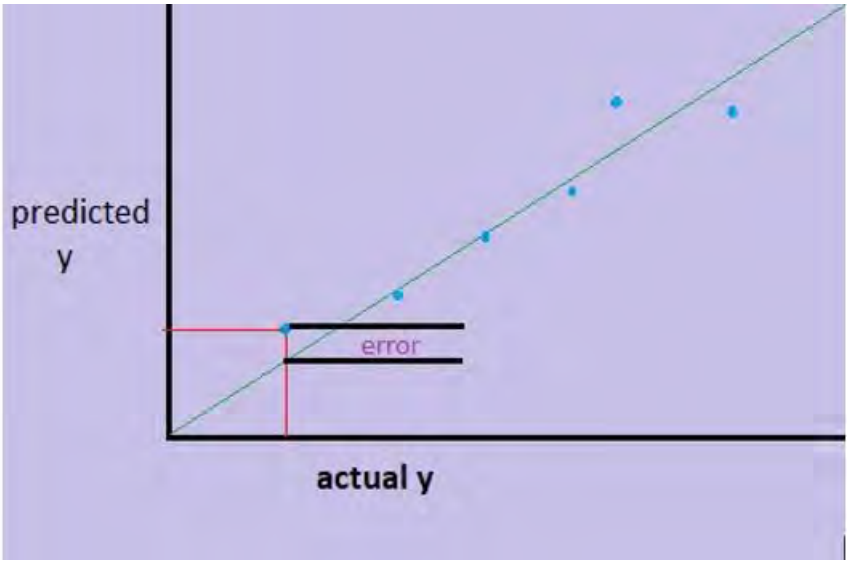


Figure 4.1: Performance Analysis

We have made a website where learners can learn Bengali sign language. Below we attached the picture of our website. The Home page contains some options. If the user want to learn from beginning then he need to press "learn sign language from the begging". He also can learn alphabets (vowel, consonant), number, some commonly used word, sentences etc.

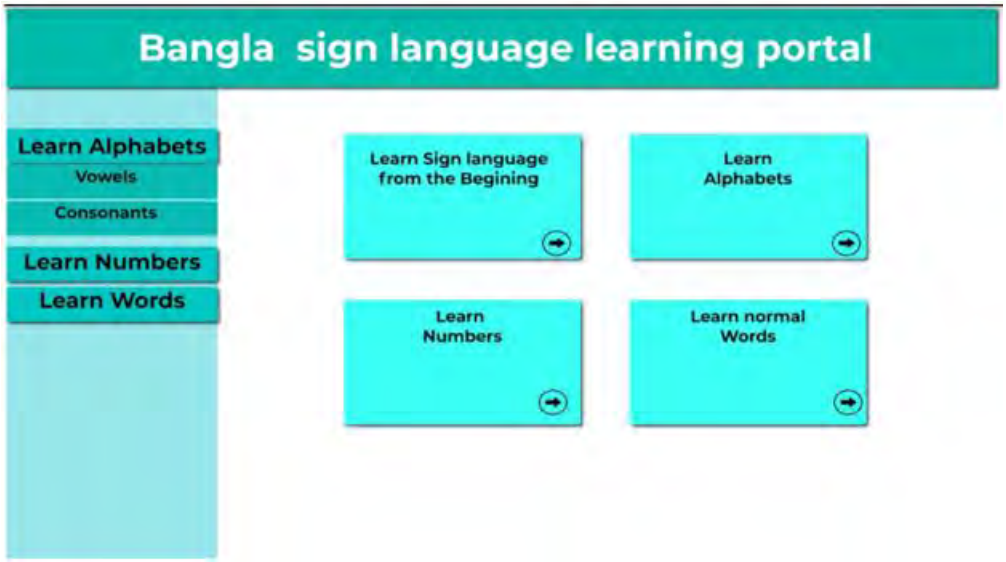


Figure 4.2: Home page of the website

At first the learner has to select what he wants to learn. If the learner wants to learn the first letter of Bengali sign language then he need to press that and then image will be appeared like below picture. Then if he wants to practice then he need to hit the practice button.

Figure 4.3: System shows sign language to user

After that it shows how the user imitates the sign language. If the user's output is very close to the model's one then it prints "good". Otherwise it will show "try again"

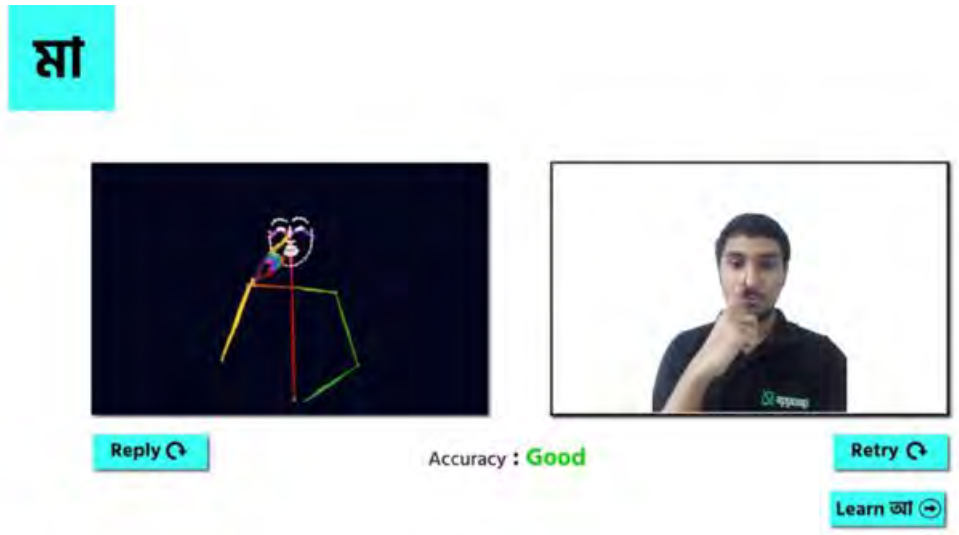


Figure 4.4: User trying with the website

For static data, the images are stored in pixel value like below:

label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	pixel10
6	149	149	150	150	150	151	151	150	151	
5	126	128	131	132	133	134	135	135	136	
10	85	88	92	96	105	123	135	143	147	
0	203	205	207	206	207	209	210	209	210	
3	188	191	193	195	199	201	202	203	203	
21	72	79	87	101	115	124	131	135	139	
10	93	100	112	118	123	127	131	133	136	
14	177	177	177	177	177	178	179	179	178	
3	191	194	196	198	201	203	204	205	205	
7	171	172	172	173	173	173	173	173	172	
8	212	212	213	212	214	213	213	213	214	
8	187	186	187	186	188	187	187	187	188	
21	128	131	133	135	137	139	140	142	145	
12	178	179	181	183	184	183	183	184	185	

Figure 4.5: Pixels of images

Here we have shown the training and test feature matrix shape

```
Training set feature matrix shape: (1381, 64, 64)
Training set classification matrix shape: (1381, 10)
Testing set feature matrix shape: (681, 64, 64)
Testing set classification matrix shape: (681, 10)
```

Figure 4.6: Train and test feature matrix shape

Then our model will be run the model with and we are given a learning rate 0.0001

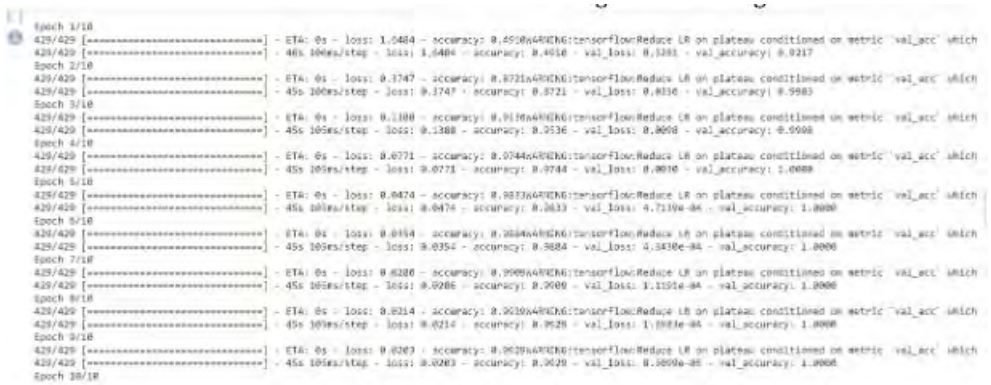


Figure 4.7: Epoch of the model after running CNN

Then our model's train and test accuracy will be 0.988 and 0.775 respectively.

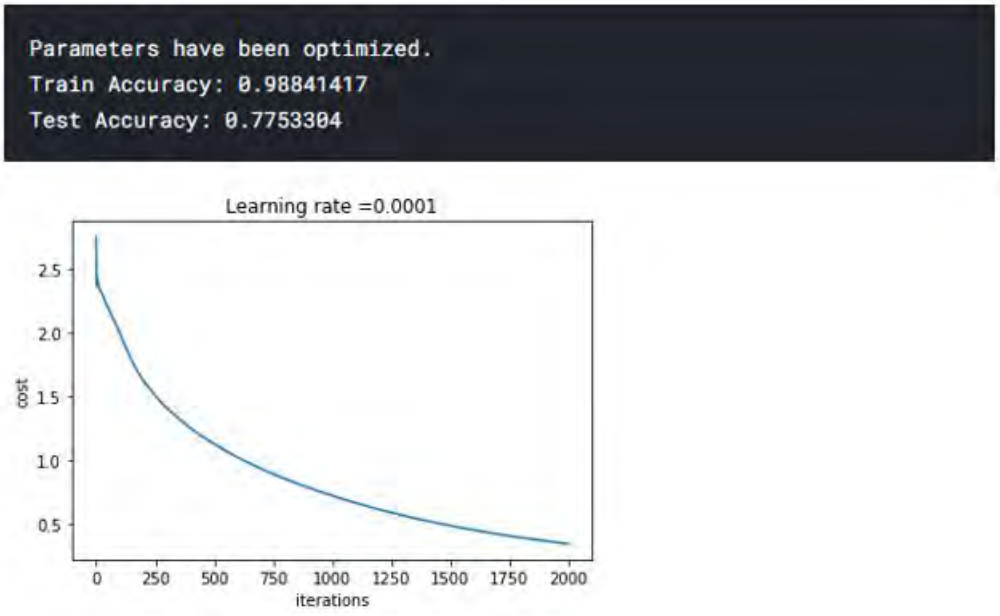


Figure 4.8: Learning rate

If the following picture is showed to our model then the answer will be

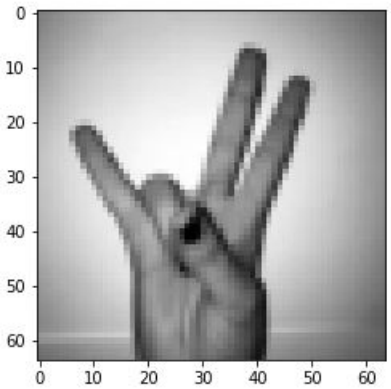


Figure 4.9: Test of sign language "7"

```
The image represents a 7 in sign-language
```

Figure 4.10: Prediction of “7”

Below we add the confusion matrix of 25 Alphabets.

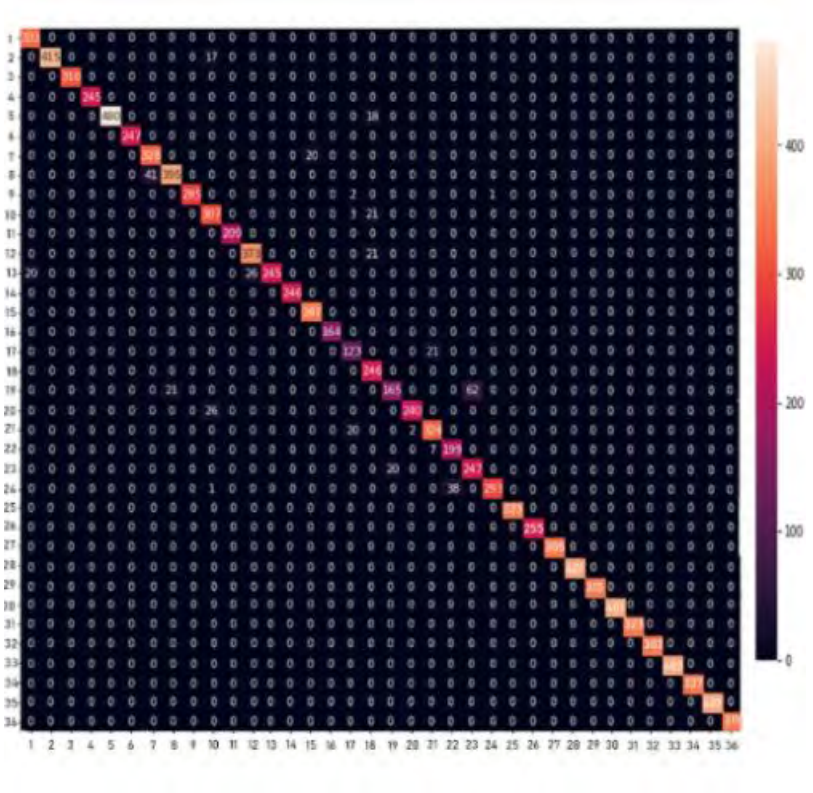


Figure 4.11: Confusion matrix of alphabets

4.2 Results

We had two models for two type of gestures. By separating them, we were able to achieve good results from our models. Our static model is able to generate 89% accuracy. On the other hand our dynamic model has 84% categorical accuracy on our training dataset and 83.2% accuracy on data that the model has not seen before. To test our dynamic model even further, we took false positive and false negative matric into account. Our model has an F score of 0.971. The early stage model has 85 percent accuracy and F score is 0.79.

If we use the images of out dataset then we get following outputs.

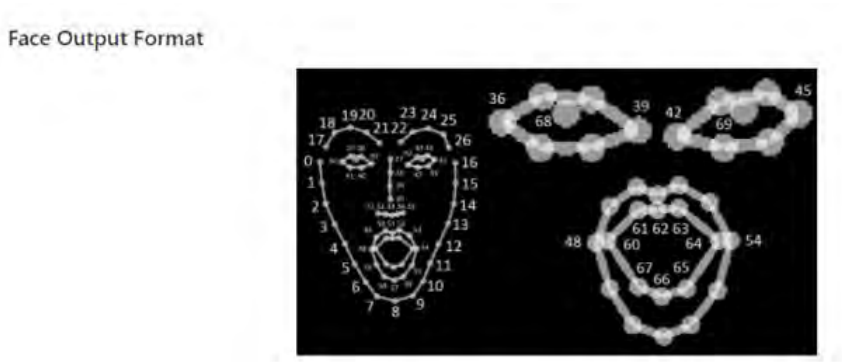


Figure 4.12: face key point

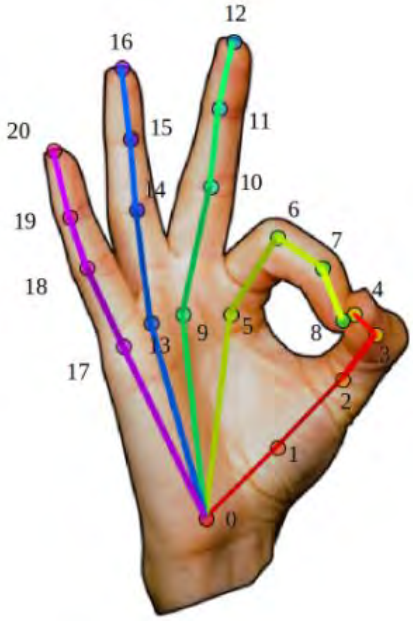


Figure 4.13: Hand's key point

The accuracy of our model is

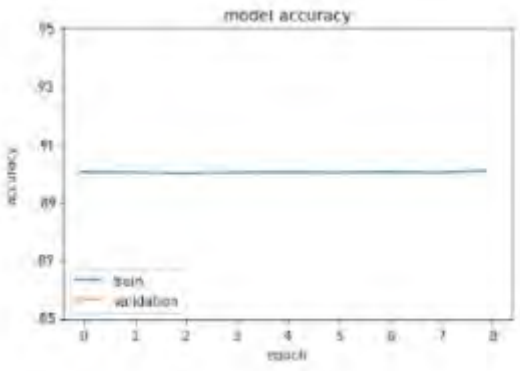


Figure 4.14: Model accuracy

Here is the loss of our model.

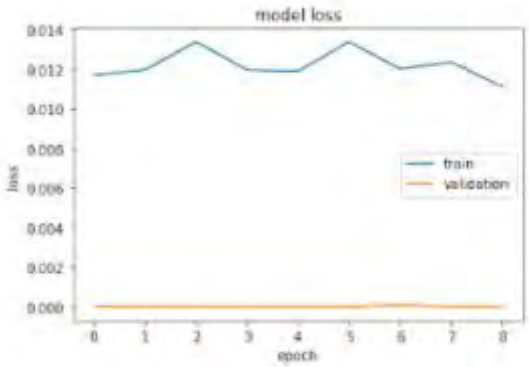


Figure 4.15: Model loss

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Our main aim is to make sign language learning easy as there are so many people uses the sign language but it is tough to communicate in that as simple gesture can change the entire meaning. Though many people can learn sign language from videos but they do not know whether they are doing it right or wrong. But in our work we will be creating the 3D model which will teach the Bangla sign language as well as after learning when they will be imitating those it will also detect if it is being done correctly or not. Until it is correct to a certain level it will keep on notifying that the imitation is wrong. There are parents who are not very much comfortable to take their disable child to take on special school to learn sign language. So, our work will help them to learn the language from home and it is more accessible. Not only it will make learning easier but also make sure whoever is learning, learning it correctly. For static sign language, we use second hand data called "Ishara Lipi" and use CNN for creating model and predicting. Second hand data often varies from the actual data. But Ishara Lipi dataset has a huge data so we think it is worthy to use this dataset in our research. For dynamic data, we use first hand data which is created by us. Then extracted the key points from the video and use it as our dataset. We have got 96.34 percent accuracy on detecting the correct numbers, vowels and consonants.

5.2 Future Work

Natural Language processing (NLP) Techniques can be used to formulate meaningful sentence from BSL words. We do not have nearly enough data to train on, and model is designed with this in mind. A more complicated RNN would likely overfit to our training data and would not perform well on data it has never seen before.For technical difficulties in this corona pandemic, we are not able to make more dataset, we needed to use cloud from internet which is not cost effective. When the situation will be fair enough then we will implement out project in our university research lab and try to minimize cost so that everyone can easily get access of our model and use it. In future dataset for BSL can be extended further for more vocabulary.The current process uses two different model CNN and RNN , in future we'll focus on combining the two models into a single model.

Bibliography

[1] M. S. Islam, S. S. S. Mousumi, N. A. Jessan, A. S. A. Rabby, and S. A. Hossain, "Ishara-lipi: The first complete MultipurposeOpen access dataset of isolated characters for bangla sign language," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, Sep. 2018. DOI: 10.1109/icbslp.2018.8554466. [Online]. Available: https://doi.org/10.1109/icbslp.2018.8554466.

[2] R. Caruana, *Machine Learning*, vol. 28, pp. 41–75, 1997. DOI: 10.1023/a:1007379606734. [Online]. Available: https://doi.org/10.1023/a:1007379606734.

[3] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, Jan. 2019. DOI: 10.1109/tpami.2017.2781233. [Online]. Available: https://doi.org/10.1109/tpami.2017.2781233.

[4] S. S. Mukherjee and N. M. Robertson, "Deep head pose: Gaze-direction estimation in multimodal video," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2094–2107, Nov. 2015. DOI: 10.1109/tmm.2015.2482819. [Online]. Available: https://doi.org/10.1109/tmm.2015.2482819.

[5] S. Liu, Y. Yin, and S. Ostadabbas, "In-bed pose estimation: Deep learning with shallow dataset," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, pp. 1–12, 2019. DOI: 10.1109/jtehm.2019.2892970. [Online]. Available: https://doi.org/10.1109/jtehm.2019.2892970.

[6] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 663–676, Dec. 2019. DOI: 10.26599/tst.2018.9010100. [Online]. Available: https://doi.org/10.26599/tst.2018.9010100.

[7] X. Du, T. Kurmann, P.-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated multi-instrument 2-d pose estimation using fully convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 37, no. 5, pp. 1276–1287, May 2018. DOI: 10.1109/tmi.2017.2787672. [Online]. Available: https://doi.org/10.1109/tmi.2017.2787672.

[8] S. S. M. Salehi, S. Khan, D. Erdogmus, and A. Gholipour, "Real-time deep pose estimation with geodesic loss for image-to-template rigid registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 470–481, Feb. 2019. DOI: 10.1109/tmi.2018.2866442. [Online]. Available: https://doi.org/10.1109/tmi.2018.2866442.

[9] R. A. Najib, N. S. T. Ahmed, and J. H. Tushar, "Sign language conversion and training by machine learning with an iot approach," Aug. 2017.

[10] Koller, O. Zargaran, S. Ney, H. Bowden, and Richard, "Deep sign: Hybrid cnn-hmm for continuous sign language recognition," 2016.

[11] S. M. F. Hasan, T. M. Rahman, A. R. Chawdhury, and A. Bishwas, "Gesture based implementation on neural network:bengali sign language to text conversion," Aug. 2017.

[12] R. R. Rumi, S. M. Hossain, A. Shahriar, and E. Islam, "Bengali hand sign language recognition using convolutional neural network," Aug. 2019.

[13] N. N. Chawdhury and G. Kayas, "Automatic recognition of bangla sign language," Dec. 2012.

[14] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," Dec. 2018.

[15] P. Guo and Z. Miao, "Multi-person activity recognition through hierarchical and observation decomposed HMM," in *2010 IEEE International Conference on Multimedia and Expo*, IEEE, Jul. 2010. DOI: 10.1109/icme.2010.5582559. [Online]. Available: https://doi.org/10.1109/icme.2010.5582559.

[16] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016. DOI: 10.1109/cvpr.2016.217. [Online]. Available: https://doi.org/10.1109/cvpr.2016.217.

[17] M. B. Magara, S. O. Ojo, and T. Zuva, "A comparative analysis of text similarity measures and algorithms in research paper recommender systems," in *2018 Conference on Information Communications Technology and Society (ICTAS)*, IEEE, Mar. 2018. DOI: 10.1109/ictas.2018.8368766. [Online]. Available: https://doi.org/10.1109/ictas.2018.8368766.

[18] A. Madylova and S. G. Oguducu, "A taxonomy based semantic similarity of documents using the cosine measure," in *2009 24th International Symposium on Computer and Information Sciences*, IEEE, Sep. 2009. DOI: 10.1109/iscis.2009.5291865. [Online]. Available: https://doi.org/10.1109/iscis.2009.5291865.

[19] M. A, "Neural networks and back-propagation explained in a simple way," Feb. 2018.

[20] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.