BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK

by
Thasin ABEDIN (160021139)
Khondokar S. S. PROTTOY (160021165)
AYANA Moshruba (160021020)

An Undrgraduate Thesis submitted to Islamic University of Technology (IUT) in partial fulfillment for award of the degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

February, 2021

BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK

Approved by:

———————————

Safayat Bin HAKIM

Supervisor and Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Boardbazar, Gazipur-1704.

Declaration of Authorship

I, **Thasin ABEDIN** (160021139)
Khondokar S. S. PROTTOY (160021165)
AYANA Moshruba (160021020), declare that this thesis titled, "BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

ISLAMIC UNIVERSITY OF TECHNOLOGY

Abstract

Department of Electrical and Electronic Engineering

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

BANGLA SIGN LANGUAGE RECOGNITION USING CONCATENATED BdSL NETWORK

by **Thasin ABEDIN** (160021139)
Khondokar S. S. PROTTOY (160021165)
AYANA Moshruba (160021020)

Communication has always been a challenge for the deaf-mute community. So sign language is the only way of interaction for them. But the problem is that sign language is way too complex for the general mass. Keeping this in mind we propose an effective alternative tool to recognise Bangla Sign Language (BdSL) using computer vision for the people in Bangladesh. In our research we propose a novel architecture, namely "Concatenated BdSL Network" combining Convolutional Neural Network (CNN) as an "Image Network" for visual feature extraction and a pretrained "Pose Estimation Network" for extraction of the hand keypoints from hand gestures. This research will hold promising future aspects for real-time sign language interpretation.

Acknowledgements

We would like to thank our supervisor Safayat Bin Hakim for the constant support and guidelines to overcome all the difficult challenges that we faced during our thesis work. Another special thanks goes to National Centre for Special Education under the Ministry of Social Welfare for providing open sourced resources for better under standing of the Bangla Sign Language. We would also like to thank Abdul Muntakim Rafi et al. as well as Bangladesh National Federation of the Deaf (BNFD) for creating such a valuable open source dataset for Bangla sign language. Finally, we thank our family, friends and all our teachers for their constant support and motivation in making this thesis project a success.

Contents

List of Figures

List of Tables

List of Abbreviations

SVM	**S**upport **V**ector **M**achine
ANN	**A**rtificial **N**eural **N**etwork
KNN	**K-N**earest **N**eighbour
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative
CNN	**C**onvolutional **N**eural **N**etwork

Dedicated To The Hearing and Speech Impaired Community

Chapter 1

Introduction

1.1 Motivation and Goals

Due to the inability to speak or hear or both, the hearing and speaking impaired community can not use the common language of interaction. Sign Language is the only mode of communication for them. It requires both visual and manual assistance.Every country has its own sign language. It is not universal at all. There are about 300 sign languages in the world like American SL, Japanese SL, Chinese SL, French SL, Spanish SL, Indian SL etc. of which American Sign Language(ASL) is the most explored by the scientists. Sign Language is very difficult to learn and more so difficult for the common people to understand that. Thus there remains a huge gap for the hearing and speech impaired group in the field of communication. The situation is worse in Bangladesh since BdSL is way tougher than ASL which is relatively straight forward. So an automated interpretation tool is extremely necessary for the impaired class in Bangladesh to make the communication convenient between the impaired and general mass.

The concept of sign language interpretation goes way back but it only gained momentum with the progress in the field of computer vision. Since it is highly inconvenient and impractical to get the constant assistance of an interpreter, scientists have been thinking of digitising the process to make lives easier for the minority group. But sign language recognition is somewhat a very complex task. Some of the methods that have been used by the researchers for the job are Support Vector Machine (SVM), Simple Recurrent Networks(SRN), Hidden Markov Model(HMM), K-Nearest Neighbour(KNN), Artificial Neural Network(ANN), Convolutional Neural Network(CNN) etc. CNN is currently the post popular method of them [18, 17, 42, 2, 14, 10].

Unlike many other sign languages the amount of research is rather small is case of detection of BdSl of which [23, 2, 25, 14, 10, 30] are worth mentioning. BdSL has 38 symbols of which 9 are vowels and 27 are consonants. In this research we propose a novel model based on Image Network(CNN) and Pose Estimation Network to extract hand features. Pose Estimation is popular among researchers for Sign Language Detection [7, 4, 12, 21]. We used a CNN model which we trained ourselves for visual feature extraction from the images. A pre-trained hand key estimation model, Openpose is used to estimate the hand key points from the image. The outputs from both

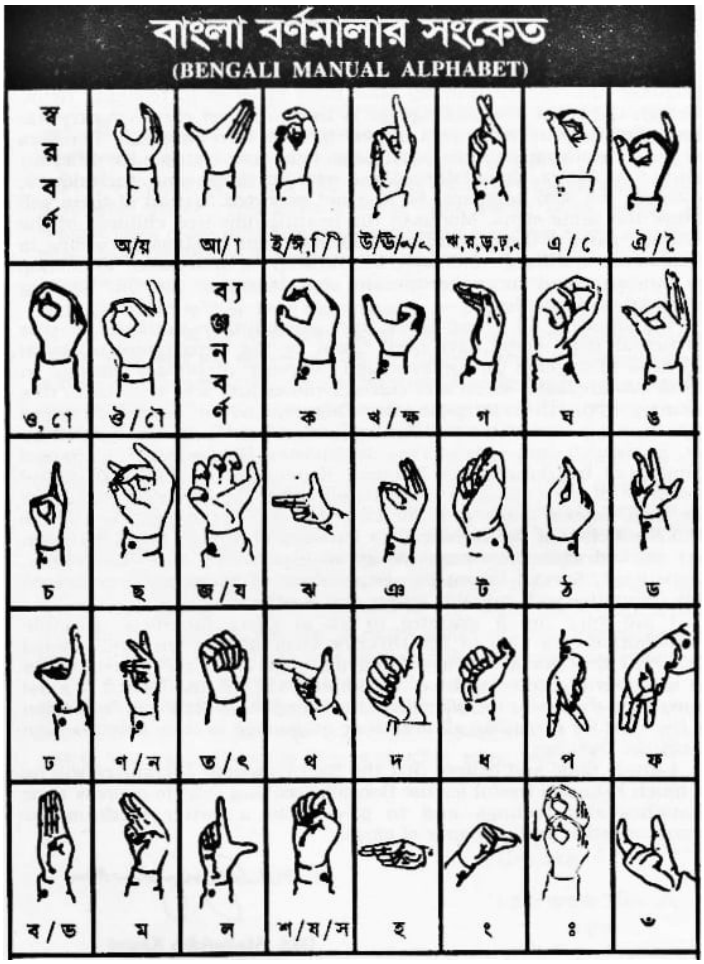

FIGURE 1.1: Bangla single hand sign language dictionary

the CNN and Openpose are then concatenated through 3 connected layers. The dataset used was the Bengali Sign Language Dataset obtained by the students of Bangladesh National Federation of the Deaf (BNFD).

1.2 Report Overview

Five chapters make up the following article. The Introduction is the first chapter in which section 1.1 elucidates the thesis' motivations and objectives and section 1.2 illustrates The project scopes of the thesis. Chapter 2 discusses the background study behind the research represented in different aspects. It further has three branches. Section 2.1 illustrates the study of different sign language recognition techniques and section 2.2 talks about the previous works in BdSL. The next section is titled as different methods of Sign Language Detection and has 5 sub sections explaining the each method briefly. Article 2.3.1, 2.3.2, 2.3.3, 2.3.4 and 2.3.5 discusses SVM, KNN,ANN,CNN and Pose Estimation Network respectively. In chapter 3, the overall methodology of our proposed architecture is discussed in details. In the first section 3.1, an in depth discussion on the proposed 'Image Network' is given along with all the required specifications. In section 3.2, details about the proposed 'Pose

Estimation Network' is given along with its working mechanism. Lastly, in section 3.3, the novel architecture 'Concatenated BdSL Network' is proposed combining the previous two sections. The data preprocessing and training details are also given in this section. The result of the research is explained in details in chapter 4. The chapter is again sub sectioned into 4 parts. In section 4.1, details on dataset is given. In section 4.2, our experimental setup is mentioned. In section 4.3, our benchmark results are included and lastly in section 4.4, in-depth result analysis is given. In chapter 5 the entire research in concluded and the future aspects and scopes of the thesis is discussed.

Chapter 2

Background Study

2.1 Different Sign Language Detection

Of the 300 sign languages existing throughout the world, not many of them have sign language recognition [33] tools . With the advancement in Artificial Intelligence, researchers have been interested to work on Sign language recognition to make life easier for the deaf-mute community. Among the other languages, most research has been performed on American Sign Language (ASL) detection. Starner et al. built real-time tool on the basis of desk and wearable computer based videos to recognise ASL [35] and again used hidden markov model in [34] which is also real-time applicable , In [44] kinetics was used for ASL recognition. Parallel hidden markov models was used in [39]. SL recognition systems have also been developing for other languages like Chinese Sign Language(CSL),Indian Sign Language(ISL), Japanese Sign Language(JSL), Arabic Sign Language, French Sign Language(FSL) etc. In [27] real time model was built for ISL. Singha et al used classification technique based on Eucliean distance which is Eigen value weighted in [32], For JSL recognition in [11] hand tracking system was built based on color, [38] showed hand feature extraction for JSL. For CSL recognition in [40] A phonemes based approach was taken. For CSL recognition a sign component based framework by using accelerometer and sEMG data [41] in [19]. These are some of the notable works in Sign Language Recognition [6].

2.2 Bangla Sign Language Detection

Bangla Sign language recognition is not yet much explored among the researchers. BdSL is actually quite unique. It is the modified form of American SL, British SL and Australian SL. Support Vector Machine (SVM) [9, 43], K-Nearest Neighbour (KNN) and Artificial Neural Network (ANN) [26] were previously the common techniques used for BdSL recognition. But in the recent times CNN is the most popular among researchers for this job. In [30] SS Shanta et al. used Scale-Invariant Feature Transform (SIFT) to extract features and CNN for detection. Abdul Muntakim Rafi and his team in [23] used VGG19 based CNN model for BdSL recognition. Md. Sanzidul Islam et al. used CNN model to recognize BdSL digits [14]. In [10] Oishee Bintey Hoque and her team used Faster R-CNN for real-time BdSL recognition. Our

proposed model introduces a novel BdSL recognition model combining both CNN and Pose estimation which is yet not done by other researchers before.

2.3 Different methods of Sign Language Detection

2.3.1 Support Vector Machine (SVM)

SVM was primarily the first choice for the sign language recognition. It is a supervised machine learning algorithm [31] typically used for classification purpose. A lot of the BdSL recognition models are based on SVM [9, 43]. SVM produces output from a set [36] of input data which is categorized [1]. The foundation of SVM is the concept of decision lines, which define decision boundaries [3] as shown in Fig.2.1. The performance is determined by the limits, and in regression, any data that is not close to the desired prediction [36] is ignored. Face recognition, bioinformatics, and image processing [1] are only a few of the applications for SVM.

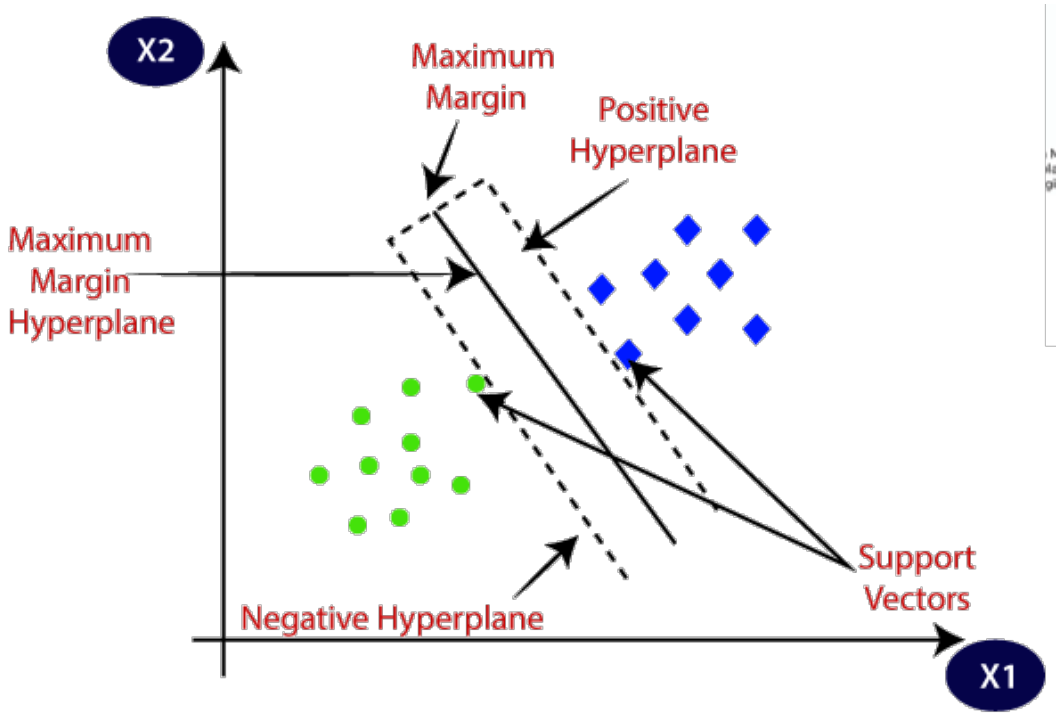

FIGURE 2.1: Support Vector Machine Algorithm [37]

2.3.2 K-Nearest Neighbour(KNN)

KNN is a very simple Machine Learning algorithm [8]. This uses no parameters for data classification. Its aim is to identify how close a data point [1] is to one of the two classes based on the collection of nearest data points. After that, it takes a set of points in space and calculates the distance between two identical points in that space using proper metrics [1] [36]. The algorithm then determines which of the training set's points are more similar

and should be taken into account when selecting the class for predicting a new observation [36] by selecting the k closest data points to that observation and assigning the most communal class among the classes [36] like in Fig.2.2.Thus, beside a new sample, a positive integer is set for k [1], and the k entries in the database that are closest to the new sample are selected and eventually the most common classification is found. KNN was used earlier for BdSL recognition [26]

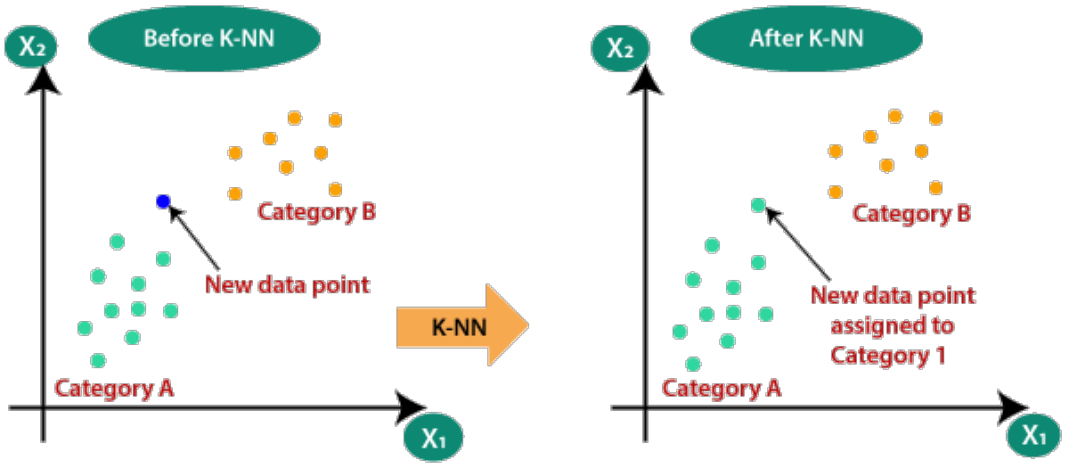

FIGURE 2.2: K-Nearest Neighbour Algorithm [15]

2.3.3 Artificial Neural Network(ANN)

ANN is a common machine learning algorithm that has the ability to recognize patterns.The system's basic architecture consists of an input layer that receives a multidimensional vector as input, which is then delivered to the hidden layer, followed by an output layer as depicted in Fig.2.3.The recognition decision is made in the hidden layer, and if there are several hidden layers, it is referred to as deep learning [1] [16] .The strength of an ANN is that it can be used to generalize problems; the machine collects data and trains on a series of patterns. This training will continue until the appropriate weight values are found, adjusting the patterns, and the machine learns the characteristics of these patterns. ANN is one of the smartest algorithms for BdSL recognition.

2.3.4 Convolutional Neural Network(CNN)

Convolutional Neural Networks is a type of deep learning that is primarily used for classifying images, clustering images based on their similarity, and performing object recognition within views.Convolutional, pooling and fully-connected layer are the three layers that make up CNN's architecture as shown in Fig.2.4 . CNN has two main parts of execution, convolution

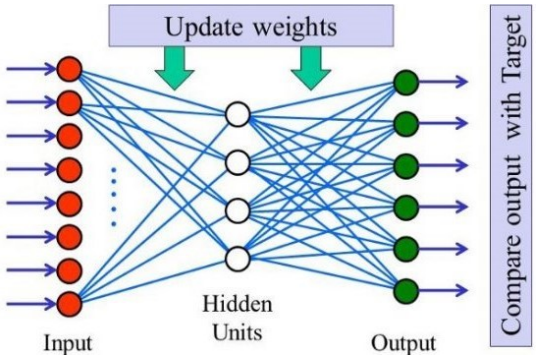

FIGURE 2.3: Artificial Neural Network Algorithm [29]

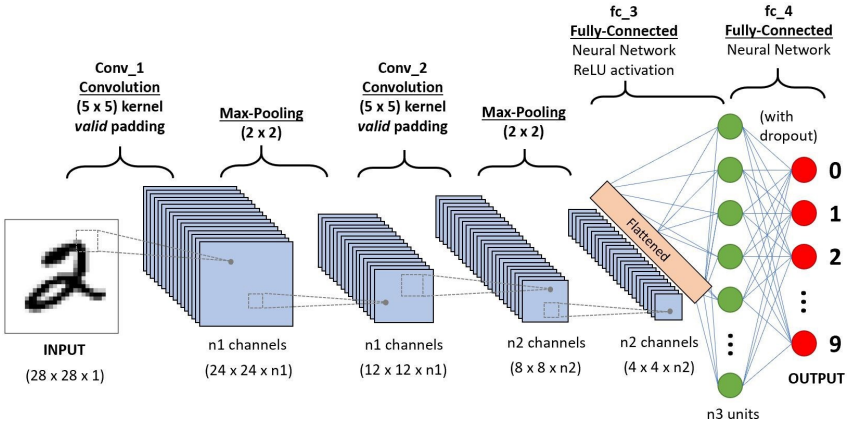

FIGURE 2.4: Convolutional Neural Network Algorithm [28]

and sampling. A trainable filter is applied to the input image in the convolutional layer, which is convolved into a feature image in each layer known as a feature map, and then a bias is applied. Detailed explanation is added in art 3.1.1. In the recent times CNN is the most popular algorithm used for BdSL [30, 23, 14, 10] In our research we built our architecture combining CNN trained ourselves and pose estimation network.

2.3.5 Pose Estimation Network

Pose estimation is the process of estimating a person's pose from a picture or video by estimating the spatial positions of key body joints using a machine learning model [22]. So when an image input is fed to the model, it will give keypoint information as output. A component ID is used to index the keypoints found, with a confidence score ranging from 0.0 to 1.0 as seen in Fig.2.5. The confidence score indicates the possibility of finding a keypoint in that location. Pose estimation is widely used in other sign language detection for feature extraction of hand gestures [7, 4, 12, 21] but it has not yet been used for BdSL recognition. We used a Pose Estimation Network (Open Pose) in our architecture which gave a better accuracy score. In BdSL, some of the symbols are very similar to each other, for example, symbol of ঊ,ঊ is identical to ড, ঙ is very similar to ঞ etc. as shown in Fig.4.6 So it is a very hard job to

distinguish the symbols from each other and recognise them correctly. Pose Estimation help extract features more accurately which helped us achieve an improved score.

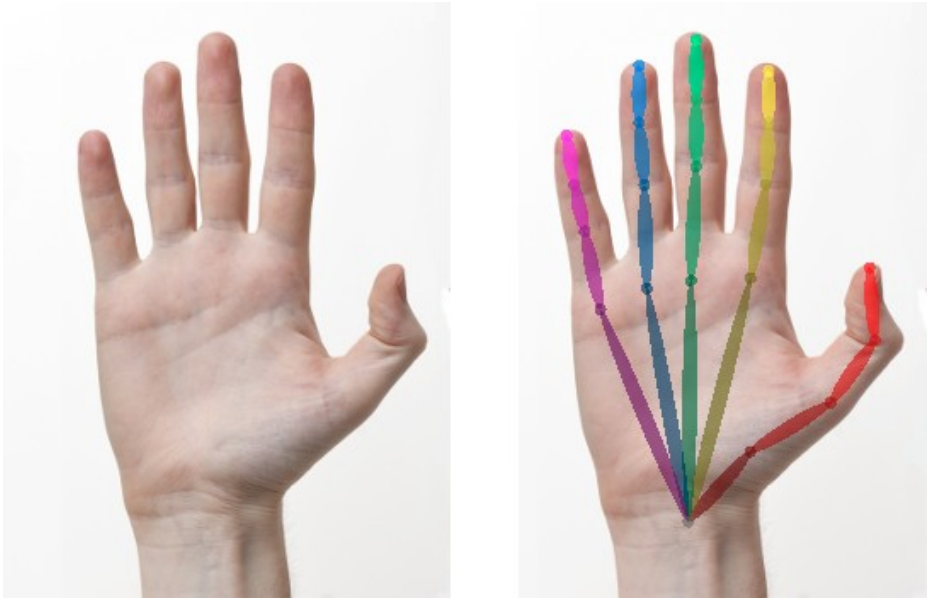

FIGURE 2.5: Hand Keypoints indexed by Pose Estimation [20]

Chapter 3

Methodology

There are several methods to recognise sign languages from images. Historically, for image recognition tasks, convolutional neural networks have provided better results compared to machine learning models. In this chapter, our main objective is to propose a deep learning based BdSL recognition architecture that aids in dealing with the complexities of the bangla sign language. Later on, in the forthcoming chapter, we analyze the performance of our novel model in respect to the existing deep learning models for bangla sign recognition to come up to a conclusion.

The method of our work falls under the category of supervised machine learning as majority of the practical work are done in this method. In supervised machine learning techniques, there is the presence of a labelled dataset and an algorithm learns based on that dataset, which is provided with an answer key to evaluate its training performance. Our proposed architecture contains of two seperate networks that are used to extract seperate features from the images to make the task of sign language detection easier. These two networks are as follows:

1. Image Network
2. Pose Estimation Network

After the two networks are done extracting seperate features from the two two individual networks, we design a mechanism to use features from both the networks. This proposed novel architecture is named as **Concatenated Bangla Sign Language(BdSL) Network.**

Each of the sections of our architecture are discussed in details in this chapter to get a better understanding of our proposed method of work.

3.1 Image Network

The main task of the image network is to extract visual features from pixel images and learn to recognize visual patterns that aid in recognizing the bangla sign language symbols. In order to extract image features, as mentioned in the literature review, previous study show the use of both machine learning and deep learning approaches. The studies also suggest that the deep learning models to perform much better than traditional machine learning approaches [30, 23, 14, 10] As a result, we use a convolutional neural network as the image network in our proposed architecture.

3.1.1 Convolutional Neural Network

A Convolution Neural Network(CNN) is one kind of deep learning technique that can extract information and learn visual patterns from images and differentiate between the images. The CNN takes an input, undergoes training by imposing weights based on feature importance of images and develops a method to recognise unique images based on its pattern recognition technique. The components of a CNN are discussed below:

- Input Layer: *The input images of the CNN are passed through the input layer. Before feeding the CNN with the input images, the input images need to be pre-processed according to the requirements to generate good results.*

- Convolutional Layer: *In order to extract features from images, a CNN uses filters to do the convolutional operation by the help of convolutional layer. The main purpose of the convolution operation is to gain both low and high level image features. In the initial convolution layers, the CNN obtains low level features such as color, edges of objects, its color gradients etc. As the CNN layers progress more and more, they help to extract the high level features like the shape of the object and give a better intuition to the CNN to get a complete idea of the object. So the convolutional layer helps to by generating feature maps of both low and high level image information in order for the network to learn visual patterns for recognition.*

- Pooling Layer: *The main task of the pooling layers is to reduce the dimension of the input visual feature that is extracted to make the process of learning easier for the neural network and minimize the computational power required to undergo training. There are mainly two types of pooling layers, namely max pooling and average pooling. In our proposed architecture, we have used the max pooling layers.*
  The max pooling layers helps to reduce dimensionality and suppress noise by taking the largest element from the rectified feature map generated by the convolution layer. [45]

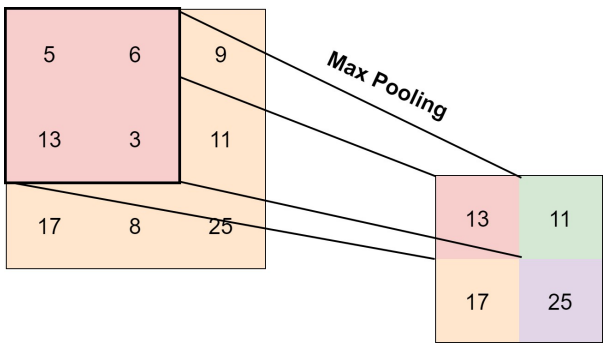

FIGURE 3.1: MaxPooling

- Fully Connected Layer: *After the task of the convolution and pooling layers are completed, the extracted information are passed through fully connected*

layers. A fully connected layers is a kind of convolution layer that has kernel size 1X1. Before feeding the extracted information into a fully connected layer, the information needs to be flattened into a one dimensional vector shape.
So the fully connected layers help to learn non-linear combinations of high-level features generated by the output of convolution layer.

- Activation Function: *An activation function is a very important component for a CNN. Based on a set of inputs to a node, it decides whether node output should be activated or not. In our proposed architecture, we mostly used the rectified linear unit (ReLu) activation function as it performs better than other activation functions. It is because it does not activate all the neurons at the same time and makes the back propagation calculation system much more easier. We have also used the eLu in some nodes that allow very small portion of negative numbers along with the positive values to activate the nodes. In the final layer, we used softmax activation to output a probability distribution of all the classes.*

- Output Layer: *The output layer of a CNN is a fully connected layer that is responsible to produce the probability distribution of all the classes present in the existing dataset. In order to get the probabilities, we assign the neuron number of the output layer equal to the number of classes in our dataset. As mentioned earlier, we use softmax activation function in this layer. Hence the output layer processes to map the input data to a vector whose elements sum up to one and generate a probability distribution to suggest probability of different classes.*

3.1.2 Proposed CNN Image Network for BdSL Recognition

In most of the relevant literature for Bangla Sign language recognition, pre-trained CNN models have been used. It is very difficult to train such models from the scratch due to lack of enough data. It also requires a lot of computational power to train a CNN from the scratch.

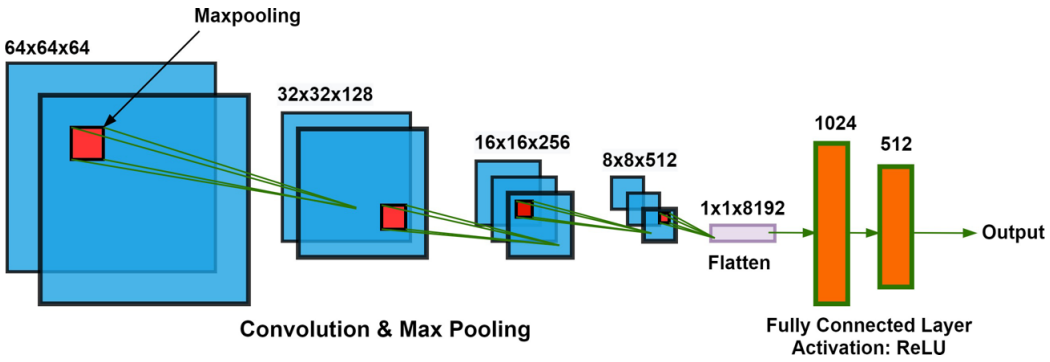

FIGURE 3.2: Image Network Architecture

However, for our proposed image network, we have trained the CNN model from the scratch. While designing the CNN from scratch, we considered the architecture of the VGG - 19 and made necessary adjustments.

TABLE 3.1: Our Image Network Architecture

Layer	Shape	Count/Value
Convolution	64x64x64	2
Convolution	32x32x128	2
Convolution	16x16x256	3
Convolution	8x8x512	3
MaxPooling	2x2	4
Dropout	-	0.2
Flatten	1x1x8192	1
Dense	1024	1
Dense	512	1

In order to make the adjustments, we have performed hyperparameter tuning of our CNN model and found the best CNN architecture suitable as the 'Image Network' for the task of BdSL recognition. The details of our image network is given in Table 3.1.

So, as we can see from the figure, the proposed image network consists of 10 convolution layers with batch normalization for each layer. It also consists of 10 ReLu activation layers, 4 max pooling layers together with a single output and input layer. The convolution layers perform convolution operation as mentioned in the previous subsection to generate "feature maps". These "feature maps" are passed to the next layers of CNN. The ReLu activation layer proves to be the best choice for our image network as it converges faster than other activation functions by not activating all the neurons at the same time. Another important reason to choose ReLu is its capability to remain unsaturated at the positive regions. After the activation layer, each layer is followed by a batch normalization to standardize the inputs and make the learning process stable. This also results in reducing the training time which is very important for us as we have trained our image network from scratch. The max pooling layers further down sample the feature maps to highlight the most prominent features to be considered only. Finally, the output of the image network is flattened to a shape of 1x1x8196.

3.2 Pose Estimation Network

In pose estimation, localization of keypoints are generated from images or videos. Pose estimation is defined as the search of a specific posture in space among all the postures that are clearly defined in the problem statement. Pose estimation can be used to estimate both 2D pose with (x,y) coordinates or 3D pose with (x,y,z) coordinates. A pose estimation network has widespread use in various use cases ranging from action recognition, people tracking, animation etc.

For the purpose of Bangla sign language recognition, we have also used a pose estimation network inorder to extract the information of hand posture while different Bangla symbols are shown. This helps the overall model to

get a better idea of the hand gesture with more precise accuracy. The pose estimation network that we used in our architecture is known as 'Openpose' [5]. A brief detail about the openpose network is given in order to get a better understanding of how OpenPose works.

3.2.1 OpenPose

OpenPose was the first model to come up with real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images. It is one of the best libraries for pose estimation and keypoints detection ranging from the detection of hand, foot, bone joints, and face. OpenPose is a bottom up approach system and uses a non parametric representation known as Part Affinity Fields(pafs). These part affinity fields helps the OpenPose network to correlate with segments of the human body in the images.

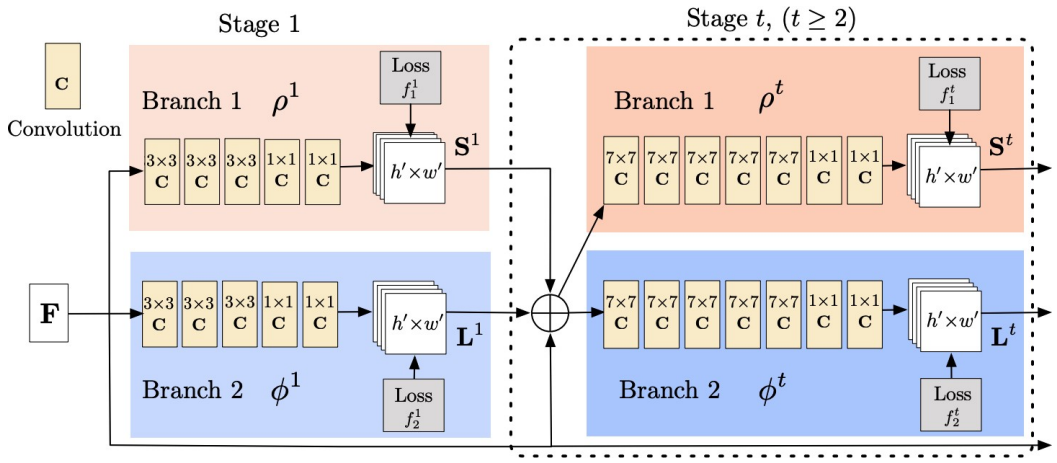

FIGURE 3.3: Openpose architecture

There are many features of OpenPose library that can be used in various scenarios. Some of the most significant features include:

- Keypoint detection of 2D multi person in real time
- Keypoint detection of 3D multi person in real time
- Posture Tracking

The working pipeline of OpenPose is very unique. Firstly, the OpenPose contains a "two branch multistage" CNN where the input images are to be fed. This CNN produces two seperate outputs. One of branch is tasked with prediction the confidence maps for different parts of the body including eyes, hands etc. On the otherhand, another branch that is at the bottom is tasked with predicting the affinity fields. The affinity fields correlate with different parts of the human body so that they can associate these different parts of the body with the input image. Lastly, after the affinity fields and confidence maps undergo greedy inference, OpenPose gives 2D keypoints for the people in the image as output of the pose estimation network.

FIGURE 3.4: Output from proposed "Pose Estimation Network"

3.2.2 Proposed Pose Estimation Network for BdSL Recognition

Although OpenPose can detect 135 keypoints in total from the human body, in case of BdSL recognition, we need to know the keypoints of the hands only. It is because the dataset that we have used for this task contains the image of human hands only. Due to this reason, in our proposed architecture, we have used OpenPose for detecting hand keypoints only as our 'Pose Estimation Network'. The primary purpose of the pose estimation network is to estimate the hand keypoints from the images. From each of the input images, the pose estimation network predicts 21 keypoints to extract the pose information from each of the bangla sign language symbols. The 21 keypoints relatively cover the whole of a single hand and gives a general estimation of the hand pose as shown in the figure below:

It is to be noted that to train a pose estimation network from scratch requires huge amount of data as well as high computational power. On top of that, OpenPose is one of the best pose estimation model for detecting human body posture. For these reasons, we use the pretrained hand keypoint estimation model from Openpose in our proposed 'Pose Estimation Model'. Finally our pose estimation model takes images as input and gives an output of shape 21 X 2. This final output that is obtained from the OpenPose is flattened to a shape of 1 X 42 in a similar way like we did for the 'Image Network'.

3.3 Concatenated BdSL Network

The task of recognizing Bangla sign language is very complex as it contains many symbols and the difference between the symbols is very subtle. For example, by a subtle change in the pose of the hands, the symbol changes from one letter to another. This adds to the complicacy of Bangla sign language recognition and can be considered one of the major difficulties of deep learning models to recognize the symbols with greater accuracy. Keeping this problem in mind, we propose our novel architecture for Bangla sign language recognition that is termed as 'Concatenated BdSL Network'.

The Concatenated BdSL Network is formed by combining 'Image Network' and 'Pose Estimation Network'. As discussed earlier, the features extracted by the image network and pose estimation network are different. However, both the features are complimentary to one another. While the image networks extracts visual information to get the overall idea of the shape of the hand image, the pose estimation network learns to associate with the different keypoints from the hand images to get a better understanding of the hand posture. Taking advantage of both these complimentary features extracted from the two networks, our proposed architecture 'Concatenated BdSL Network' tries to tackle the complicacy in Bangla sign language mentioned earlier. In order to do so, the final flattened output of both the image and pose estimation network are further passed through two fully connected

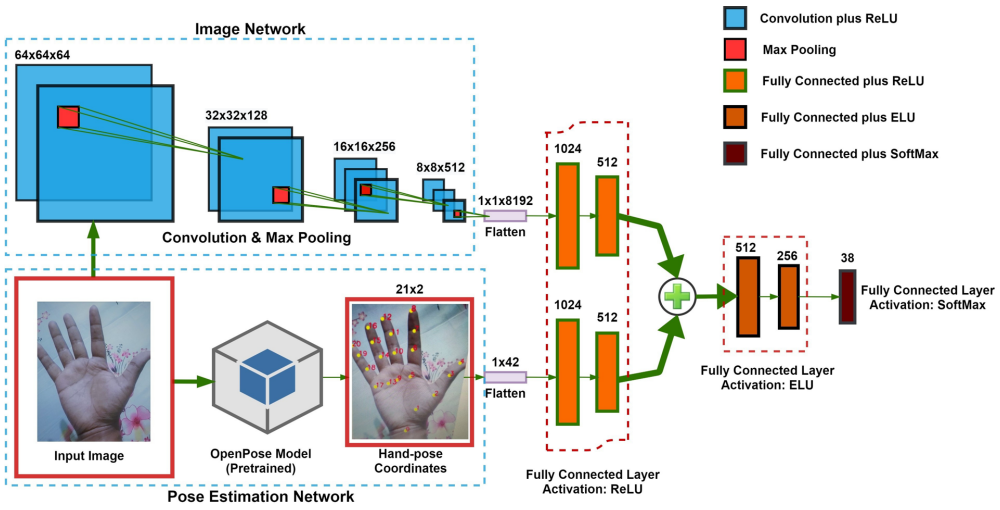

FIGURE 3.5: Concatenated BdSL Network

layers with ReLu activation function and concatenated together. This concatenation is done to combine the complimentary features that are extracted from both the image network and pose estimation network. Hence, we term our proposed architecture as Concatenated BdSL Network. Later on, the newly obtained features by concatenation are further passed through 3 more fully connected layers, where the first two layers have "eLu" activation and the last one has "softmax" activation function to generate a "probability distribution" of all the symbols.

3.3.1 Data Preprocessing

In our proposed architecture, the image network and pose estimation network requires input images seperately. Infact, the format in which the images are fed to both the networks are also different. The image network requires the images to be converted in grayscale format. On the other hand, for the pose estimation network, the input images are required to be kept in RGB format as OpenPose needs it to be like that while feeding the input. Apart from that, in the preprocessing stage, both the input images need to be resized into 64 X 64 size and normalized by the highest level (255). These are done for faster training purpose keeping our computational constraints in mind.

3.3.2 Training and Evaluation Method

At the beginning of the task, we divide the dataset in train and test set. Details about the dataset and the train test split are discussed in the next chapter. Before the start of training, we do the data preprocessing steps that were mentioned earlier and feed the data according to the requirements of our image network and pose estimation network. After doing the preprocessing steps, the preprocessed input of the image network is converted into a

numpy file. During training time, this input that is converted into numpy array is fed directly as the input to the image network.

For the pose estimation network, the method to feed the input is a little bit different. At first after doing the data preprocessing, we fed that data through the pretrained OpenPose. The OpenPose generates hand keypoints output in the form of coordinate values. These generated output of hand keypoints are converted into another seperate numpy array and stored as a seperate numpy file. During training time, the output of the OpenPose that is converted into a numpy array is directly fed as to the pose estimation network as the second input of the Concatenated BdSL Network. As we are performing supervised learning, during training time we also converted the labels associated with the corresponding images as a seperate numpy file so that our proposed model can learn to recognize the Bangla symbols. During the time of evaluation, we also follow the same procedures starting from data preprocessing upto the conversion to two seperate numpy array which are to be fed directly to our proposed model. The only difference during evaluation time is that we do not need to give our model the labels corresponding to the image as it is the task of the model to generate the labels itself during the evauluation process.

It is also to be mentioned that during training time as cost function, we utilize cross entropy function. Inorder to reduce our cost function to a minimal value, we use a gradient descent based "Adam" optimization that has a learning rate of 0.001. This learning rate was not kept static. We update the learning rate value whenever the validation results of the model did not improve for 4 consecutive epochs using the reduce learning rate on plateau mechanism. For the training, we also initialize the weights with small numbers. In order to avoid overfitting during training, we also use early stopping mechanism to interrupt training after a certain time if we see that the validation loss has not improved upto a certain number of epochs.

Chapter 4

Results

Our main goal was to justify adding a pose estimation network alongside commonly used image network in hand sign recognition works. To analyse the importance of the pose estimation network, we compared our novel model with modified VGG-19 Image Network [23] as well as our own Image Network. We also experimented with commonly used machine learning architectures for this kind of tasks like SVM, ANN and KNN.

4.1 Dataset

We used the Bengali Sign Language Dataset which was collected by [23] from the students of Bangladesh National Federation of the Deaf (BNFD). Single hand sign from 'Bangla Sign Language Dictionary' was followed during the creation of this dataset. The dataset was collected from 193 male and 143 female persons. Among them, 42 persons have hearing imparity. Their age range is of 12 to 30. This is a single hand bangla sign language dataset which contains all 38 bangla sign language letters without any mathematical symbols showed in Fig.4.1.

This dataset contains 11061 train and 1520 test images. They are inside 38 differently labeled folder. Total size of this dataset is only 188MB as the images are of low resolution (224 by 224 pixels). Each image contains only the hand symbol with static background. We splited the traing dataset into training and validation with 9959 and 1102 images respectively following previous work [23]. Due to lack of GPU, we trained both of our Image only model and Concatenated BdSL model for 30 epochs.

4.2 Experimental Setup

We conducted our experiments under identical computational environment —-

- Framework: *Tensorflow 2.2*
- Platform: *Google Colaboratory*
- CPU: *1-core allocated Intel Xeon processo 2.2GHz*

FIGURE 4.1: Sample images of our used dataset

- Physical Memory: 12.72GB
- Virtual Machine Storage: 107.77GB
- GPU: *None*

4.3 Benchmark Tests

We used some commonly used techniques such as classification accuracy using test sets, confusion matrix, precision, recall, f1-score to evaluate our classification models.

4.3.1 Classification Accuracy

Classification accuracy gives the percentage of correct predictions. The equation is given below:

TABLE 4.1: Comparison of Classification Accuracy score on test data

Model	Test Accuracy
Concatenated BDSL Network	**91.51**
Image only Network	90
Modified VGG-19 Image Network	89.6
Support Vector Machine	22.89

$$Classification\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

where,

TP = **T**rue **P**ositive

TN = **T**rue **N**egative

FP = **F**alse **P**ositive

FN = **F**alse **N**egative

This benchmark score works best with datasets with evenly distributed samples in all classes. So, we need F1 score which we also included in our benchmark tests. Our test set have 40 samples of images in each individual classes. As it is perfectly balanced, our test accuracy score is perfectly acceptable. Without knowing the details of the work, it is pretty good benchmark score nonetheless.

As we can see from the Table-4.1, our novel concatenated BdSL network got higher score in test accuracy than our image only network, support vector machine and modified VGG-19 image network on the same dataset. It is necessary to mention that, previously modified VGG-19 image network had the best score on the same test data.

4.3.2 Confusion Matrix

Confusion matrix is often used to visually show the performance of a classification model. This is to show whether the model is confusing two classes. It is shown in Table-**??**. Here, only true positives and true negatives are desired outcomes. So, the more the confusion matrix is diagonal, the more the classification model is accurate. We can see the accuracy of individual classes from this. It is also used to calculate precision and recall.

As we can see from Fig.4.4, each cell got score between 0 to 40 as there are only 40 labels available for each class. The values in the diagonal cells are correct predictions. The values of upper triangles are false negatives and the

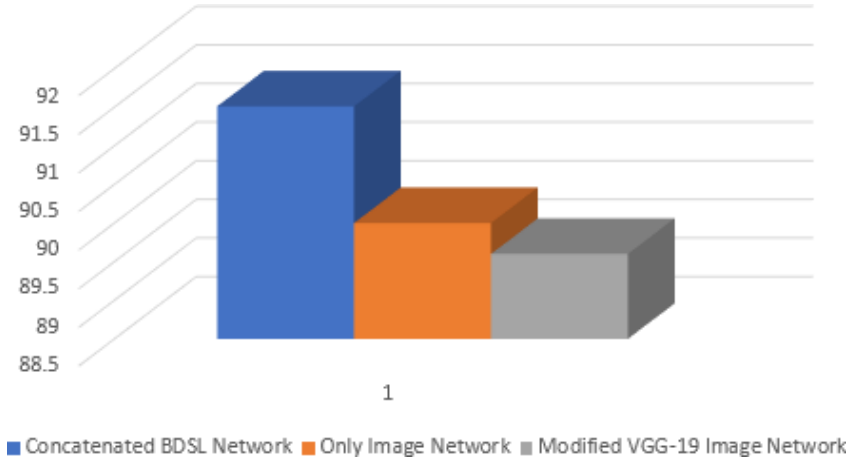

FIGURE 4.2: Comparison chart of our novel model and different image network model

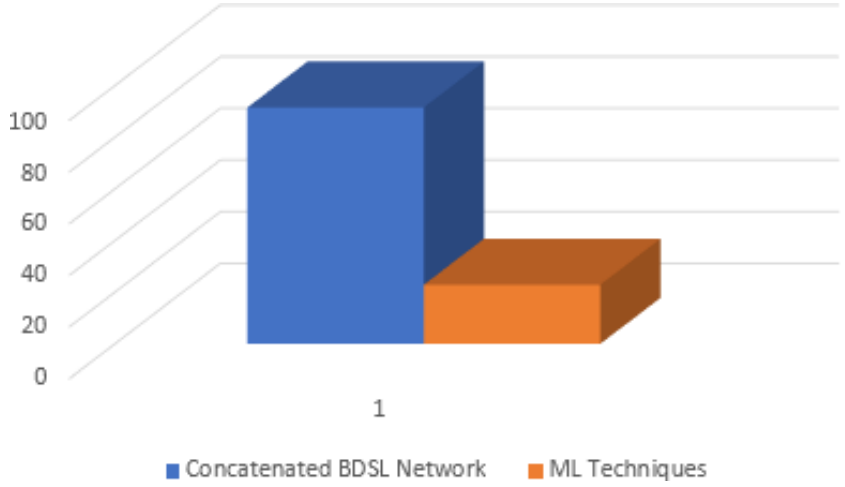

FIGURE 4.3: Comparison chart of our novel model and Machine learning techniques

values of bottom triangles are false positives. Our confusion matrix is not 100 percent diagonal as there are some misclassifications among classes. Major wrong predicted classes are highlighted using red boxes.

4.3.3 Recall, Precision and F-1 Score

Precision and recall metrics are the next step from the classic classification accuracy. They allow us to get more specific understanding of our classifier model. They are very important metric to use while dealing with uneven distribution of data.

I **Recall**

Recall score indicates how good a model is at predicting positive classes.

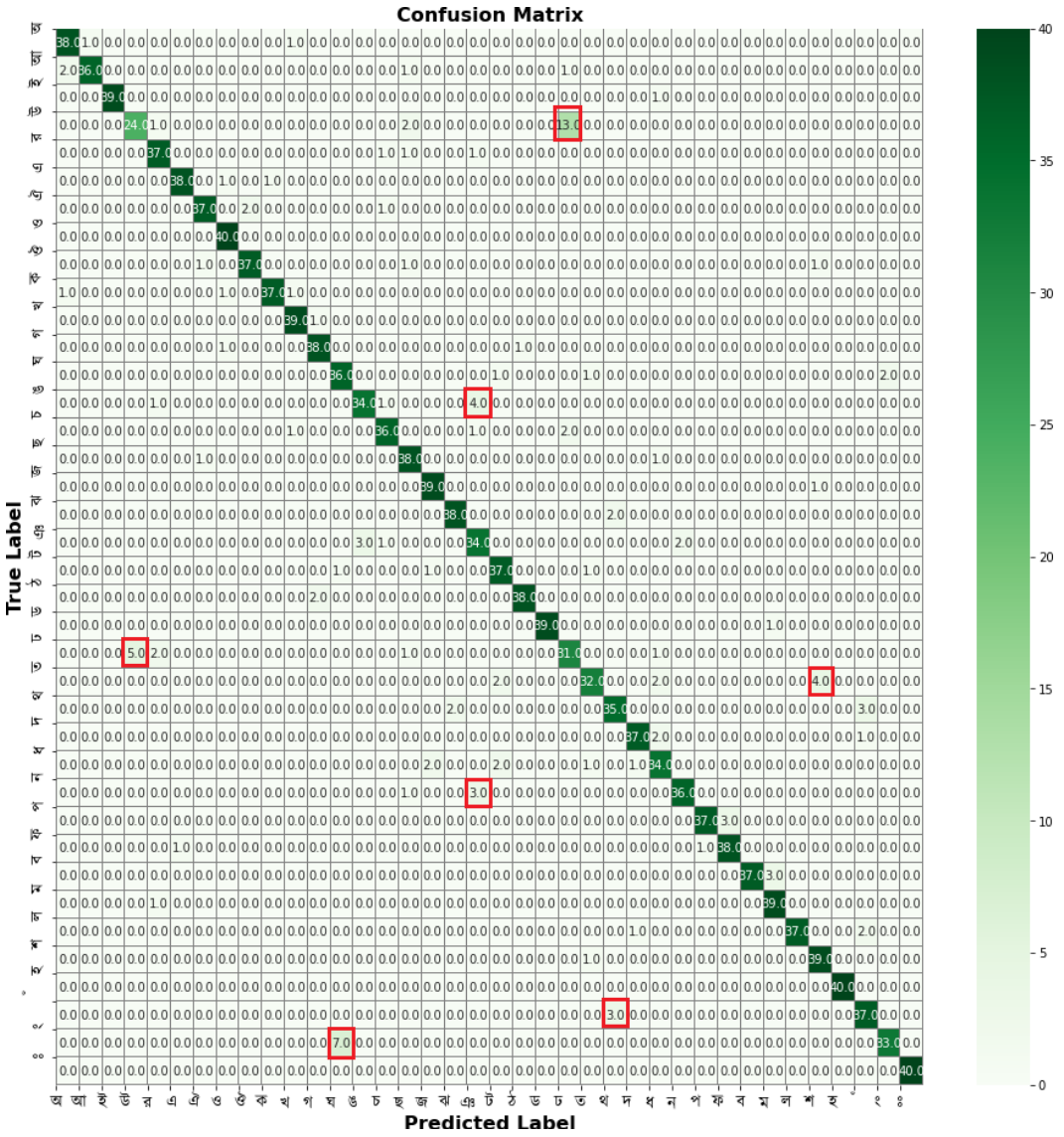

FIGURE 4.4: Generated confusion matrix using 40 test samples in each individual classes

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

II Precision

Precision score indicates how many positive predictions are true.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Increasing the recall automatically decreases the precision. So, one can't maximize both of them and prioritize one based on one's goal and task. F1-Score includes both recall and precisión into a single number.

III F1-Score

"F1-Score" is a better choice for calculating accuracy of a classification model. It takes both recall and precision into account as we can see from the equation below:

$$F1\_Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4.4}$$

Its value ranges from 0 to 1 and the higher is the better. It solves the issue of uneven distribution of data between classes completely.

In Fig.4.5, we can see results of individual classes as well as the number of data we used to evaluate our model. Our model got average score of 0.92 which is very good in these type of tasks.

4.4 Result Analysis

We evaluated every model on the testing set comprised of 1520 signs. All 38 classes have same amount of 40 samples from 40 different people. Our novel model handly beaten our main competitor across the board which can be seen in Fig.4.2. We tested some machine learning techniques using the same dataset but it is very clear from the Fig.4.3 that machine learning techniques performs very poorly in image classification tasks compare to widely used convolutional neural networks. We also put our novel model against our own Image network which is our proposed architecture excluding the pose estimation network. We did it to show the justification for adding the pose estimation network. We can see a bit better performance over an already good performing Image network in Fig.4.2.

Every hand sign consists of unique positioning of five fingers and palm. Most signs are not drastically different from each other. With just visual feature extraction used in classic image network, we cannot know the relative position of a certain finger, its pose and interaction with other fingers and palm. But using pose estimation network, we can label each finger joint with its unique number. Thus each finger has its own four unique numbers. With this type of feature extraction, we can accurately identify finger positions which is needed in this type of classification. By combining both visual and positional information, our novel model achieved even higher precision. We couldn't compare our work with other similar works due to vastly different datasets like two handed bangla sign language [10, 24, 13] were used. Our novel model achieved "state-of-the-art" result on this data.

Despite being a very good classification model, our confusion matrix is not 100 percent diagonal as there are some misclassifications among classes. We can see the same scenario in f1-score as well. It is pretty clear that some signs are very close in looks and they are quite indistinguishable even with human eyes. So it is not that easy for our classification model either.

From Fig.4.6 we can see that, these eight hand gesture are nearly identical to each other. It is possible that some of these signs were wrongly labelled in the dataset. So it is also a reason for our model to get poor results in these signs.

	Precision	Recall	F1-Score	Support
অ	0.93	0.95	0.94	40
আ	0.97	0.90	0.94	40
ই	1.00	0.97	0.99	40
উ	0.83	0.60	0.70	40
ঋ	0.88	0.93	0.90	40
এ	0.97	0.95	0.96	40
ঐ	0.95	0.93	0.94	40
ও	0.93	1.00	0.96	40
ঔ	0.95	0.93	0.94	40
ক	0.97	0.93	0.95	40
খ	0.93	0.97	0.95	40
গ	0.93	0.95	0.94	40
ঘ	0.82	0.90	0.86	40
ঙ	0.92	0.85	0.88	40
চ	0.90	0.90	0.90	40
ছ	0.84	0.95	0.89	40
জ	0.93	0.97	0.95	40
ঝ	0.95	0.95	0.95	40
ঞ	0.79	0.85	0.82	40
ট	0.88	0.93	0.90	40
ঠ	0.97	0.95	0.96	40
ড	1.00	0.97	0.99	40
ঢ	0.66	0.78	0.71	40
ত	0.89	0.80	0.84	40
থ	0.88	0.88	0.88	40
দ	0.95	0.93	0.94	40
ধ	0.83	0.85	0.84	40
ন	0.95	0.90	0.92	40
প	0.97	0.93	0.95	40
ফ	0.93	0.95	0.94	40
ব	1.00	0.93	0.96	40
ম	0.91	0.97	0.94	40
ল	1.00	0.93	0.96	40
শ	0.87	0.97	0.92	40
হ	1.00	1.00	1.00	40
◌ঁ	0.86	0.93	0.89	40
◌ং	0.94	0.82	0.88	40
◌ঃ	1.00	1.00	1.00	40
Accuracy			0.92	1520
Macro avg	0.92	0.92	0.92	1520
Weighted avg	0.92	0.92	0.92	1520

FIGURE 4.5: F1-score on test data

TABLE 4.2: Performance Comparison

	Concatenated BDSL Network	**Modified VGG-19 Image Network**
Training	**98.67**	97.68
Validation	**95.28**	91.52
Testing	**91.51**	89.6
Image Size	**64x64**	224x224
Epoch	**30**	100
GPU	**No**	Yes

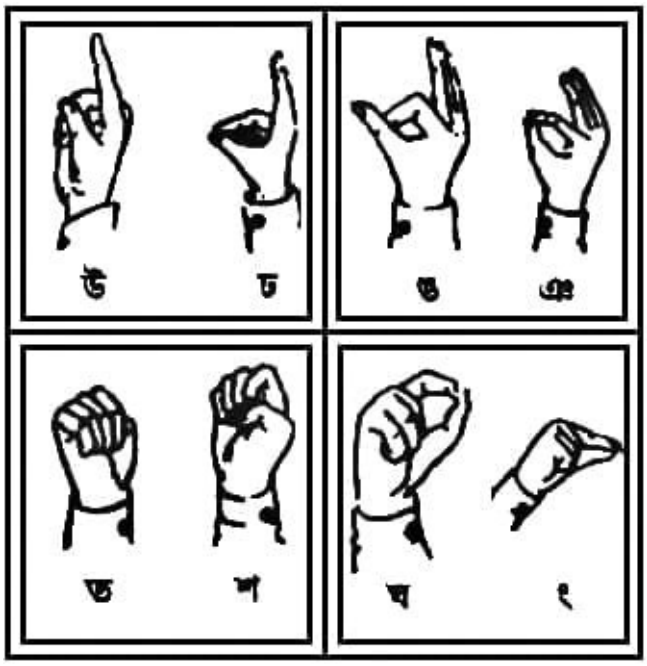

FIGURE 4.6: 8 nearly identical bangla single hand signs

Chapter 5

Conclusion and Future Study

Life of the minority group is never easy specially here in Bangladesh. We do not have the ability to drastically improve their situation but through this research we aim to make a difference even if that's little. Our research generates an automatic tool for BdSL alphabets for single hand gestures. Our architecture is a concatenated network combining CNN and Pose Estimation Network which enabled us to get a satisfactory result with the available computational capacity. But there are promising scopes for this research. With a greater computational power this model will thrive even more to get higher accuracy.

Our model takes still images as input but with still images it is not going to help the deaf-mute community properly. Because without real time implementation the tool won't be practically realisable. With further work on the subject our goal is to make our model work on real time videos. After that, continuous sentences must be identified and mapped to the appropriate spoken grammar. Lastly, in order to understand beyond the literal sense of a sign and to minimize recognition errors, facial expressions and other sign parameters must be taken into account. Also, to make communication a two-way operation, text-to-sign and sign-to-text systems must be merged into one system. So we need to collect a huge amount of real time continuous BdSL videos to make sufficient large training datasets. Then we can implement our proposed model to detect continuous BdSL by training it with computationally efficient resources.

And finally we have to make a model suitable for commercial use. In the future, continuing research in this field may lead to a variety of other applications, such as sign language guides or dictionaries, as well as making it easier for the deaf and mute to browse the web or send emails.

Bibliography

[1] Rawan A Al Rashid Agha, Muhammed N Sefer, and Polla Fattah. "A comprehensive study on sign languages recognition systems using (SVM, KNN, CNN and ANN)". In: *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*. 2018, pp. 1–6.

[2] Fahmid Nasif Arko et al. "Bangla sign language interpretation using image processing". PhD thesis. BRAC University, 2017.

[3] Sebastiano Battiato et al. *Image Analysis and Processing-ICIAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part I*. Vol. 10484. Springer, 2017.

[4] Manas Kamal Bhuyan, Mithun Kumar Kar, and Debanga Raj Neog. "Hand pose identification from monocular image for sign language recognition". In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE. 2011, pp. 378–383.

[5] Zhe Cao et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2021), pp. 172–186.

[6] George Caridakis, Stylianos Asteriadis, and Kostas Karpouzis. "Non-manual cues in automatic sign language recognition". In: *Personal and ubiquitous computing* 18.1 (2014), pp. 37–46.

[7] James Charles et al. "Automatic and efficient human pose estimation for sign language videos". In: *International Journal of Computer Vision* 110.1 (2014), pp. 70–90.

[8] Swagatam Das et al. *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*. Vol. 404. Springer, 2015.

[9] Muttaki Hasan, Tanvir Hossain Sajib, and Mrinmoy Dey. "A machine learning based approach for the detection and recognition of Bangla sign language". In: *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*. IEEE. 2016, pp. 1–5.

[10] Oishee Bintey Hoque et al. "Real time bangladeshi sign language detection using faster r-cnn". In: *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE. 2018, pp. 1–6.

[11] Kazuyuki Imagawa, Shan Lu, and Seiji Igi. "Color-based hands tracking system for sign language recognition". In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE. 1998, pp. 462–467.

[12] Jason Isaacs and Simon Foo. "Hand pose estimation for american sign language recognition". In: *Thirty-Sixth Southeastern Symposium on System Theory, 2004. Proceedings of the*. IEEE. 2004, pp. 132–136.

[13] Md Islam et al. "Recognition Bangla Sign Language using Convolutional Neural Network". In: Sept. 2019, pp. 1–6. DOI: 10.1109/3ICT.2019.8910301.

[14] Sanzidul Islam et al. "A potent model to recognize bangla sign language digits using convolutional neural network". In: *Procedia computer science* 143 (2018), pp. 611–618.

[15] *K-Nearest Neighbor(KNN) Algorithm for Machine Learning*. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning?fbclid=IwAR0aGwUDx_5Kj2N2-Ucv-9cf5ck-jLnIr-TFTga5u7yHj16jG2LM83GAQrQ.

[16] Nahua Kang. "Multi-Layer Neural Networks with Sigmoid Function—Deep Learning for Rookies (2)". In: *Towards Data Science* (2017).

[17] Oscar Koller et al. "Deep sign: hybrid CNN-HMM for continuous sign language recognition". In: *Proceedings of the British Machine Vision Conference 2016*. 2016.

[18] Oscar Koller et al. "Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos". In: *IEEE transactions on pattern analysis and machine intelligence* (2019).

[19] Yun Li et al. "A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data". In: *IEEE transactions on biomedical engineering* 59.10 (2012), pp. 2695–2704.

[20] Marcelo Ortega. *Training a Hand Detector like the OpenPose one in Tensorflow*. https://medium.com/@apofeniaco/training-a-hand-detector-like-the-openpose-one-in-tensorflow-45c5177d6679 Access Date: Feb 27, 2021

[21] Maria Parelli et al. "Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos". In: *European Conference on Computer Vision*. Springer. 2020, pp. 249–263.

[22] *Pose Estimation*. https://www.tensorflow.org/lite/examples/pose_estimation/overview

[23] Abdul Muntakim Rafi et al. "Image-based Bengali Sign Language Alphabet Recognition for Deaf and Dumb Community". In: *2019 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE. 2019, pp. 1–7.

[24] Muhammad Rahaman et al. "Bangla Language Modeling Algorithm For Automatic Recognition of Hand-Sign-Spelled Bangla Sign Language". In: *Frontiers of Computer Science (electronic)* 14 (Aug. 2018). DOI: 10.1007/s11704-018-7253-3.

[25] Muhammad Aminur Rahaman. "Computer vision based Bangla sign language recognition". PhD thesis. University of Dhaka, 2018.

[26] Muhammad Aminur Rahaman et al. "Real-time computer vision-based Bengali sign language recognition". In: *2014 17th International Conference on Computer and Information Technology (ICCIT)*. IEEE. 2014, pp. 192–197.

[27] P Subha Rajam and G Balakrishnan. "Real time Indian sign language recognition system to aid deaf-dumb people". In: *2011 IEEE 13th international conference on communication technology*. IEEE. 2011, pp. 737–742.

[28] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53?fbclid=IwAR2krVrepGRcAstJc_eHBcrI2VP3mnCxZbrFRnb_4ZnbB1EmCbTLSPqGsso. Feb 27, 2021.

[29] Lee Schlenker. *Artificial Neural Networks: Man vs Machine*. https://groupfuturista.com/blog/artificial-neural-networks-man-vs-machine/?fbclid=IwAR1kDq1ZA6ZA1J5YvMdLzbohs8smApCnJ3hW9fyGeO8cF092XZjJBhaOQmQ. Access Date: Feb 27, 2021

[30] Shirin Sultana Shanta, Saif Taifur Anwar, and Md Rayhanul Kabir. "Bangla sign language detection using sift and cnn". In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE. 2018, pp. 1–6.

[31] Andrzej Sieminski et al. *Modern approaches for intelligent information and database systems*. Vol. 769. Springer, 2018.

[32] Joyeeta Singha and Karen Das. "Indian sign language recognition using eigen value weighted Euclidean distance based classification technique". In: *arXiv preprint arXiv:1303.0634* (2013).

[33] Tomáš Sixta et al. "Fairface challenge at ECCV 2020: analyzing bias in face recognition". In: *European Conference on Computer Vision*. Springer. 2020, pp. 463–481.

[34] Thad Starner and Alex Pentland. "Real-time american sign language recognition from video using hidden markov models". In: *Motion-based recognition*. Springer, 1997, pp. 227–243.

[35] Thad Starner, Joshua Weaver, and Alex Pentland. "Real-time american sign language recognition using desk and wearable computer based video". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.12 (1998), pp. 1371–1375.

[36] *SUE Academics*. https://academics.su.edu.krd/#1.

[37] *Support Vector Machine Algorithm*. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm?fbclid=IwAR2LtSPbU4hRDZMFBGcuZD2OfyWM_4GXmm915UDLv6m2fI4kAjMO15sQgxQ.

[38] Nobuhiko Tanibata, Nobutaka Shimada, and Yoshiaki Shirai. "Extraction of hand features for recognition of sign language words". In: *International conference on vision interface*. 2002, pp. 391–398.

[39] Christian Vogler and Dimitris Metaxas. “Parallel hidden markov models for american sign language recognition”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 1. IEEE. 1999, pp. 116–122.

[40] Chunli Wang, Wen Gao, and Shiguang Shan. “An approach based on phonemes to large vocabulary Chinese sign language recognition”. In: *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*. IEEE. 2002, pp. 411–416.

[41] Shengjing Wei et al. “A component-based vocabulary-extensible sign language gesture recognition framework”. In: *Sensors* 16.4 (2016), p. 556.

[42] Su Yang and Qing Zhu. “Continuous Chinese sign language recognition with CNN-LSTM”. In: *Ninth International Conference on Digital Image Processing (ICDIP 2017)*. Vol. 10420. International Society for Optics and Photonics. 2017, 104200F.

[43] Farhad Yasir et al. “Sift based approach on bangla sign language recognition”. In: *2015 IEEE 8th International Workshop on Computational Intelligence and Applications (IWCIA)*. IEEE. 2015, pp. 35–39.

[44] Zahoor Zafrulla et al. “American sign language recognition with the kinect”. In: *Proceedings of the 13th international conference on multimodal interfaces*. 2011, pp. 279–286.

[45] R. Ķēniņš. “Land Cover Classification using Very High Spatial Resolution Remote Sensing Data and Deep Learning”. In: *Latvian Journal of Physics and Technical Sciences* 57.1-2 (1Apr. 2020), pp. 71 –77.