Bengali Hand Sign Language Recognition Using Convolutional Neural Networks

by

Roisul Islam Rumi
15301033
Syed Moazzim Hossain
15301092
Ahmed Shahriar
15301119
Ekhwan Islam
15301132

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 and Engineering

Department of Computer Science and Engineering
Brac University
April 2019

© 2019. Brac University
All rights reserved.

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:

Roisul Islam Rumi 15301033	Syed Moazzim Hossain 15301092
Ahmed Shahriar 15301119	Ekhwan Islam 15301132

Approval

The thesis titled "Bengali Hand Sign Language Recognition Using Convolutional Neural Networks" submitted by

1. Roisul Islam Rumi (15301033)
2. Syed Moazzim Hossain (15301092)
3. Ahmed Shahriar (15301119)
4. Ekhwan Islam (15301132)

Of Spring, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on April 25, 2019.

Examining Committee:

Supervisor:
(Member)

Mr. Hossain Arif
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Md. Saiful Islam
Lecturer
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Md. Abdul Mottalib, PhD
Professor and Chairpersion
Department of Computer Science and Engineering
BRAC University

Abstract

Throughout the world the number of deaf and mute population is rising ever so increasingly. In particular Bangladesh has around 2.6 million individuals who aren't able to communicate with society using spoken language. Countries such as Bangladesh tend to ostracize these individuals very harshly thus creating a system that can allow them the opportunity to communicate with anyone regardless of the fact that they might know sign language is something we should pursue. Our system makes use of convolutional neural networks (CNN) to learn from the images in our dataset and detect hand signs from input images. We have made use of inception v3 and vgg16 as image recognition models to train our system with and without imagenet weights to the images. Due to the poor accuracy we saved the best weights after running the model by setting a checkpoint. It resulted in a improved accuracy. The inputs are taken from live video feed and images are extracted to be used for recognition. The system then separates the hand sign from the image and gets predicted by the model to get a Bangla alphabet as the result. After running the model on our dataset and testing it, we received an average accuracy of 99%. We wish to improve upon it as much as possible in the hopes to make deaf/mute communication with the rest of the society as effortless as possible.

Keywords: Bangla Sign Language (BSL); CNN; Deep Learning; Artificial Intelligence; Image Processing.

Dedication

We would like to dedicate this thesis to our parents, family members and friends ...

Acknowledgement

All praises are due to The Almighty Allah, The Most Gracious, The Most Merciful who blessed us to be able to pursue an undergraduate degree at BRAC University.

Our deepest gratitude goes to our Supervisor, Mr. Hossain Arif and our Co-Supervisor, Md. Saiful Islam for supporting us, guiding us and helping us every step of the way to complete this thesis.

We would also like to thank Ms. Umme Hani of the CSE Department, International Islamic University, Chittagong for providing us with the preliminary dataset which was very important for our work.

And finally, we would like to thank our parents, family members and friends for their unwavering support and love. Our gratitude goes out to all of them.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN Artificial Neural Network

ASL American Sign Language

$BdSL$ Bangladeshi Sign Language

BSL Bangla Sign Language

$BSLA$ Bangla Sign Language Anthology

CNN Convolutional Neural Networks

DCT Discrete Cosine Transform

HSV Hue Saturation Value

ISL Indian Sign Language

KNN K-Nearest Neighbor

LCF Local Cosine Feature

MLP Multi-Layer Perceptron

NCL Negative Correlation Learning

NNE Neural Network Ensemble

PCA Principal Component Analysis

RGB Red Green Blue

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Motivation and Goals

Humans communicate with each other by speaking or writing in a structured conventional way using their own native languages or other languages' alphabets. However, deaf and mute people that cannot hear or speak have no other choice but to live their daily lives without any other feasible means. So, sign language becomes the only feasible mean for their communication. Furthermore, there being no standard sign language internationally different nations use their own type of gestures. Bangla Sign Language Anthology (BSLA) is the standard when it comes to Bangladesh. Bangla Sign Language (BSL) has a structure that does not match with the sign languages of other countries. In BSL, for expressing alphabets generally both hands are used. This is why an interpreter is really essential to understand what they are trying to convey. Sign Language Recognition (SLR) has become one of the most prominent fields of research in current times and with the aid of innovative technologies, it can help bridge the communication gap that deaf people have with normal people.

The purpose of our work is to contribute in automatic SLR. We are focusing on detecting the numbers and alphabets of BSL. There are two main steps of our framework. First being the process of extracting features from the images which will result in one or more feature vectors which are also called descriptors. These feature vectors can be both local or it could be global as well. In our project we are mostly taking into account the vectors, and the second step will be the classification based on the feature vectors between different signs. For general image classification tasks, several types of Convolutional Neural Networks (CNNs) [10], [13] are used as they are considered to be a robust tool when it comes to supervised feature learning and classification, so we are using CNN to implement our system.

1.2 Project Scope

Digitising sign language and using it as a program to help make the communication gap between deaf and normal people narrower. Other countries have already made their sign language digital but BSLA have not yet been digitised. Few others before us have worked on sign recognition on the full BSLA but we are doing that as well as Bangla text conversion at the same time. By making this program we are insuring that as few as possible children feel as outcasts by society due to communication difficulties.

1.3 Overview

The project started with intensive research into multiple different methods and devices related to sign language detection for both Bangla sign language and other sign languages, mainly American and Indian. For our project we have used different approaches and methodologies to fulfill different purposes. We have created own images for our dataset and are used for pre-processing. For the training of the inception v3 model, the pre-processed images are used. After training, different batches of images that are not from our dataset were used for validation and testing.

The other research projects that we have taken inspiration from have used other methods and are to some extent different from our approach. But, the method that we have used are better and more efficient than the others. If we were to talk about pre-processing, our model just needed the dataset of images while the other methods had to do image re-sizing and then RGB to black and white conversion and so and so forth. They had to put in a lot of effort and time, but ours did not require such a hassle, so ours is more efficient. Moreover, our approach is also unique because the papers that we researched did not make use of the inception v3 model as we have used in our system.

Chapter 2

Literature Review

Primarily, a lot of research and projects have been conducted on the topic of sign language recognition and detecting hand signs, be it using just images or using gesture recognition through advanced hardware like gloves with sensors. Our primary focus is to develop a system mostly based on software and using a camera to take images and use them to detect hand signs. Moreover, various algorithms and methods have been used and discussed in these papers. For image binarization we have Otsu algorithm [14] and for image denoising there is morphological image processing [8]. If the feature set gets too large in its dimension, Principal Component Analysis [1] and Kernel PCA [4] reduces the dimensionality. For the detection of shapes and edges convex hull algorithm [3] and Canny edge detection algorithm [6] are frequently used. Furthermore, the most commonly used classifiers for such tasks include support vector machine [5] and neural networks [9]. Among the logistic regression-based classifiers, the multi-layer perceptron [2] is also considered ideal for matching features with a dataset.

2.1 History of Sign Language

Sign Language in western nation was recorded in 17th century. It was created as a method of communication using visual means. It contains gestures, mimicry, hand and finger signs. Usually sign languages can also show complete ideas or phrases other than individual words. From the mid 19th century, development in sign language was properly discussed but was mostly limited to philosophical or theoretical discussion. However, if we were to talk about the mentions of sign language then we can travel much further than the 17th century. In 360 B.C. Plato in his book Cratylus posed a very philosophical question, "Assume that we do not have a voice or tongue but wanted to point out objects to one another. Should we not, like the people that have no voice, make signs with our hands, head and the rest of the body?". During the 15th century, the Ottoman courts had a population of deaf people in the hundreds, whose job was to teach the masses sign language since it was impolite to speak in front of the sultan. In Scituate, Massachusetts around 1600 there was a large population of deaf people due to natural causes and intermarriages thus sign language became the natural form of communication. this allowed the formation of one of the first deaf school.

2.2 Artificial neural network training using fingertip position

Relative position of the fingertips in a two-dimensional space and position vectors are used to train an artificial neural network (ANN). The dataset included 518 images of hand signs taken with a high resolution camera and the pixel values of the images were used in training the ANN. The Canny Edge Detection Algorithm was used for the images to detect the edges first and then detect the position of the fingertip of each finger. Each finger is then assigned a (X,Y) value with respect to the position and the collection of (X,Y) values for five fingers represents a hand sign. The classification is done using ANN and its three layered architecture containing input, output and hidden layer. Input layer took the 10 (X,Y) values of the fingertips from both hands and the output layer matched the input with the 37 classes of hand signs, and the hidden layer was defined by the user. Back propagation was used in training the ANN with 60 percent of the data and remaining was used for testing the accuracy. These methods proved quite useful and had a detection accuracy of 99 percent. We intend to include this methodology in our system as well to get optimum results[20].

2.3 Artificial neural network and support vector machine

The Microsoft Kinect camera can be used for taking input images of hand signs and detect the joints and wrists by finding the contours. The convex hull method is used as the contour finding algorithm to extract contour features and sends them to the support vector machine (SVM) for detection. The inital RGB input image is taken and converted to a grayscale image and then to a binary threshold image, and then the convex hull algorithm detects joints, bends and angles by drawing a polygon around the hand shape. After joint detection, the SVM takes the data and compares it with the hand signs in the dataset and gives a message if there is a match. Their system had an accuracy of 84.11 percent in detecting hand signs[25].

2.4 Hand sign recognition using Gabor filter, Kernel PCA and SVM

M. A. Uddin and S. A. Chowdhury conducted a research on hand sign recognition in 2016 where they have reported that they took RGB images of hand signs and converted them to HSV and then a Gabor filter is applied to it to extract the desired features. The output obtained from this is in a higher dimension, so for the reduction in the dimension, a technique called Kernel PCA (Principal Component Analysis) is used. And for training and classification they used SVM. Initially, the images of hand signs are taken with a digital camera and then the RGB images undergo conversion, first from RGB to HSV and then from HSV to binary, which makes it easier for segmentation. Then a morphological operation is performed on the images for noise reduction in them by first performing dilation, then erosion and finally smoothing operation. Furthermore, they used image segmentation to separate the region into parts according to the vertical and horizontal lines. Moreover, the resulting binary image is resized and a Gabor filter is applied on it. The Gabor filter is useful in this case for feature extraction as it can operate properly regardless of the lighting conditions. The resulting feature vectors are too large, so to make the process easier, they used a Kernel PCA to reduce the dimensions and extract the patterns. Finally, SVM is used to train and classify the images, half of them used in training and the other half used for testing the accuracy. Their technique proved to be formidable as they were able to detect the hand signs perfectly (100 percent) in normal conditions and with significantly high accuracy in different angles (98.6 percent) and varied lighting conditions (99.5 percent) for a total accuracy of 99.5 percent[23].

2.5 Using PCA to recognize hand signs in real time

Authors S. N. Sawant and M. S. Kumbhara mention in their research paper, Real Time Sign Language Recognition using PCA, that they extracted hand sign features of Indian Sign Language using the Eigenvectors and Eigenvalues from the images and then using Principal Component Analysis (PCA) algorithm for gesture recognition and conversion to text. The images are taken with a webcam while there is a white background behind the hand for ease in processing. Segmentation and morphological filtering methods are applied for separating the hand object from the background. For segmentation, Otsu algorithm is used while the morphological filtering methods denoise the image and give a uniform contour. Here, PCA algorithm is also used to reduce dimensionality and to extract features. Then the extracted values are compared with the dataset of 260 images of 26 hand signs and corresponding matches are displayed as text[16].

2.6 American sign language (ASL) recognition in real time using leap controller

In their research paper, D. Naglot and M. Kulkarni, Real Time Sign Language Recognition using the Leap Motion Controller, they used leap motion controller which is a contactless 3D motion sensor that can do tracking and detection of the hands and fingers. A Multi-Layer Perceptron (MLP) with a neural network along with a Back Propagation (BP) algorithm builds the classification model by taking feature sets as inputs. The dataset contained 520 samples, that is 20 samples each of the 26 alphabets of ASL, and their system had an accuracy of 96.15 percent in detecting hand signs[21].

2.7 A 3D stroke based representation of sign language signs using key maximum curvature points and 3D chain codes

In the paper written by Geetha M and M. R. Kaimal (2017), they were able use a unique approach to detecting different Indian Sign characters using a stroke based identification system and they were also able to take into account facial expressions that helps in understanding for the meaning of word in its entirety. In their paper, the authors claim that by extracting only the key frames efficiently they are able to increase their overall accuracy and lower the total time complexity. Individuals sign words uniquely however the major curvatures that occur when signing are found to be very much the same thus the writers used Key Maximum Curvature Points algorithm to identify the key frames where curvature is most prominent. Once the keystrokes are recognized using Stroke Sequence Identification Algorithm they are then passed through a Hidden Markov Model which uses the Baum-Welch Training and Viterbi Algorithm for error and rectification. Finally, the whole process ends when the keystroke sequence is matched to words using the stroke decision tree[28].

2.8 Two-Handed Sign Language Recognition for Bangla Character Using Normalized Cross Correlation

In the paper written by Kaushik Deb, Helena Parvin Mony Sujan Chowdhury (2012), they developed a model of sign detection which in layman terms works by using RGB to refine the hand and wrist band sign region to get heuristic threshold values that gets filtered through a template matching technique which recognizes the signs. In the color segmentation part that occurs in the refinement process, each of the signs will have four different colors, which are red, swan, skin and black (the background). The detection algorithm is broken down into 3 major component that are color segmentation, centroid calculation and finally localizing the edge of the wrist bands[12].

Chapter 3

System Implementation

3.1 Proposed Model

The flowchart of our proposed model, shown in Fig. 3.1, initially checks if relevant images are available. If images are available, the system reads and separates the images from the video and crops out the area of interest. The area of interest is then resized to match the resolution of the training dataset. After cropping the image, it then sent to the inception v3 model. The image with the highest prediction accuracy is picked. The resulting character is shown as output.

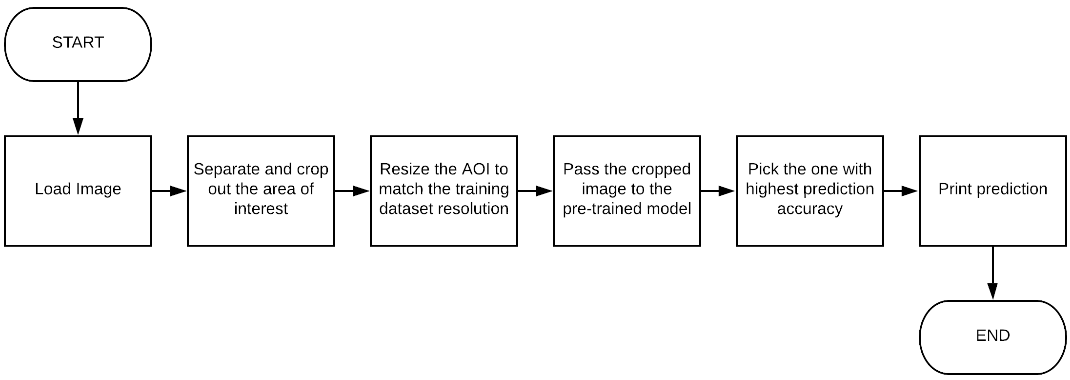

Figure 3.1: Flowchart of proposed model

3.2 Convolutional Neural Networks

A Deep Learning algorithm that takes images as input and assigns various weights or levels of importance to the features of the images and as a result will be able to differentiate between the images. As this approach requires the least amount of pre-processing, it is preferable over other classification algorithms. The pattern of this algorithm is similar to the connectivity pattern of the human brain and its working process is also inspired by it.

The input images are taken as color planes in the RGB spectrum and reduces the images into a form that makes it easier to process and aid in prediction. A 5x5x1

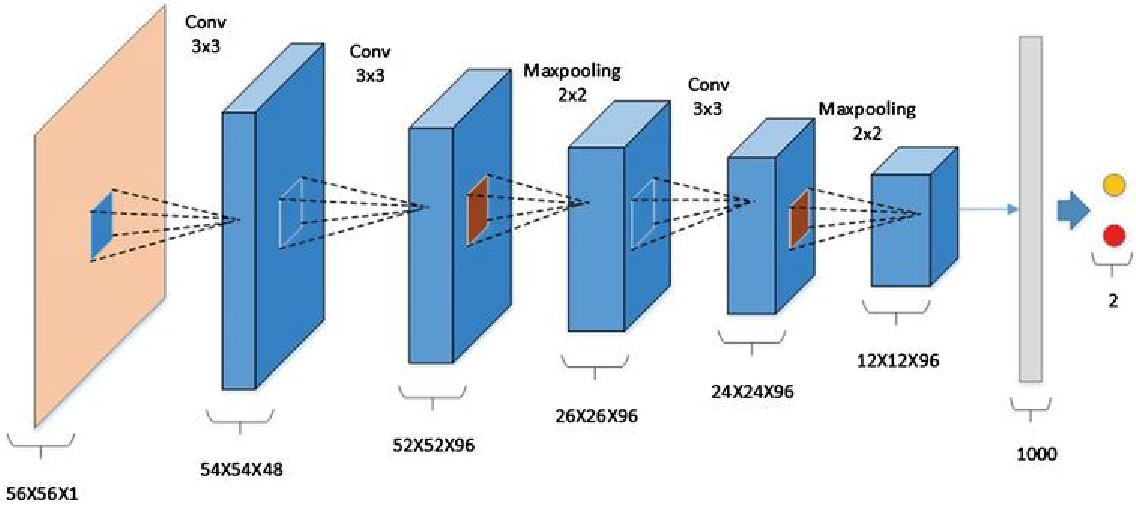

Figure 3.2: CNN model

input image can be converted to 3x3x1 using a kernel/filter, K, in the convolutional layer. The K here is 3x3x1 matrix. The kernel traverses the entire image from left to right and shifts to the lower row and repeats and does matrix multiplication along the way. This allows us to obtain the high-level features, like edges from the image.

3.3 Dataset Collection

We started the dataset collection by receiving a dataset of all the 45 classes whose image was of a single individual with black background. Some hand signs from this dataset are shown in Fig. 3.3, 3.4 and 3.5. From this initial dataset we expanded upon it by adding two more 45 image set of two other individuals. With these 3 sets done done we applied augmentation on it through scripts and other methods to further expand our dataset. In the augmentation process we had a number of parameters. These parameters were rotation angle, width shift range, height shift range, shear range, zoom range, horizontal flip and fill mode. Once the augmentation was done we had for each class 100 images for training, 200 images for testing and finally another 200 images for evaluation. For a grand total of 1,400 images for each of the 45 classes that puts our total number of images at 54,000.

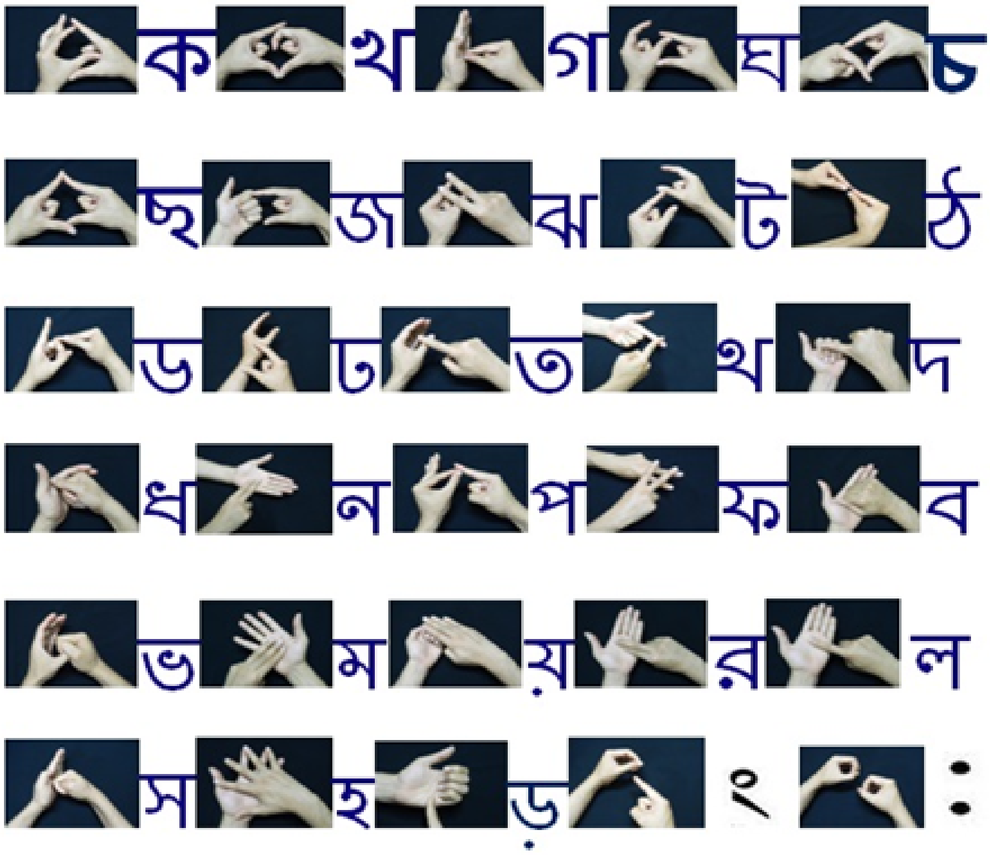

Figure 3.3: Bengali hand sign alphabets from a part of our dataset

Figure 3.4: Bengali hand signs of vowels from a part of our dataset

3.4 Inception v3

The idea behind this Convolutional Neural Network architecture started with the first iteration of the Inception model called inception v1. The idea behind v1 was to improve the utilization of the computing resource inside of the network. The way

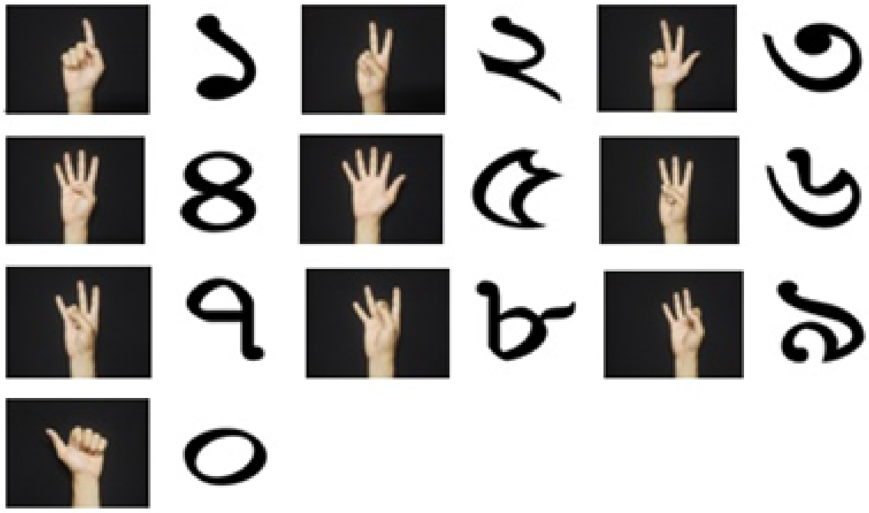

Figure 3.5: Bengali hand signs of numbers from a part of our dataset

this improvement was achieved was by increasing the width, the total number of units at any one level, and depth, the final number of levels, of the network while keeping the overall cost constant. The version-1 was 27 layers deep with 1x1, 3x3 and 5x5 filter size, which were used more for convenience rather than necessity[19].

Next comes the inception v2 and v3 as they are both just upgraded version of their previous counterpart. In v2 the authors were trying to reduce the loss of information or as they liked to call it “representational bottleneck”. Also, they wanted to improve the computational complexity such that convolution could be more efficient. These refinement were brought out by factorizing the 5x5 size into two 3x3 convolution operation. The 5x5 operation is about 2.78 times more expensive thus by using two 3x3 operation a notable increase in the speed of variation can be observed[18]. Finally the version we have implemented v3 has all the upgrades of v2 with a few addition. One of the main ones is the factorization of the 7x7 convolution also label smoothing, a type of regularizing component useful for preventing overfitting[22].

The input images are first taken and in the trained dataset and each category is split into similar number of images. Then the training labels are received from the training sets. We trained our own CNN which is the inception v3 model. The images are pre-processed for the inception v3 model. They are then split into train test validation. Feature extraction will be done from the deeper layers of the inception v3 model and classify the category of the image. The trained images are validated using the validation set. Using the trained model and trained classifiers, the label is predicted for the test set. The overall accuracy is then displayed.

Figure 3.6: Some Bengali hand signs from our extended dataset for training

3.5 VGG16

It's a convolutional network model proposed by Karen Simonyan and Andrew Zisserman (2015). One of the major benefits that has been observed about this model is that it has scored 92.7% on the top-5 test accuracy in ImageNet. The 16 in vgg16 stands for the depth of 16 weight layers that this model can go to using the 3x3 filter. By doing this it's possible to observe a significant amount of improvement[17].

3.6 VGG19

VGG19 is the more advanced as well as the more complex evolution of the VGG16 model. It is a convolutional neural network model similar to the VGG16 however the key difference in between these models is the number of layers each are employing. In the VGG16 model as denoted by the number 16 we can see that it is using 16 layers for image classification. And in the VGG19 model it uses 19 layers for image classification as such its called VGG19. Another improvement in the VGG19 is the number of object categories it contains, compared to its predecessor the VGG16.

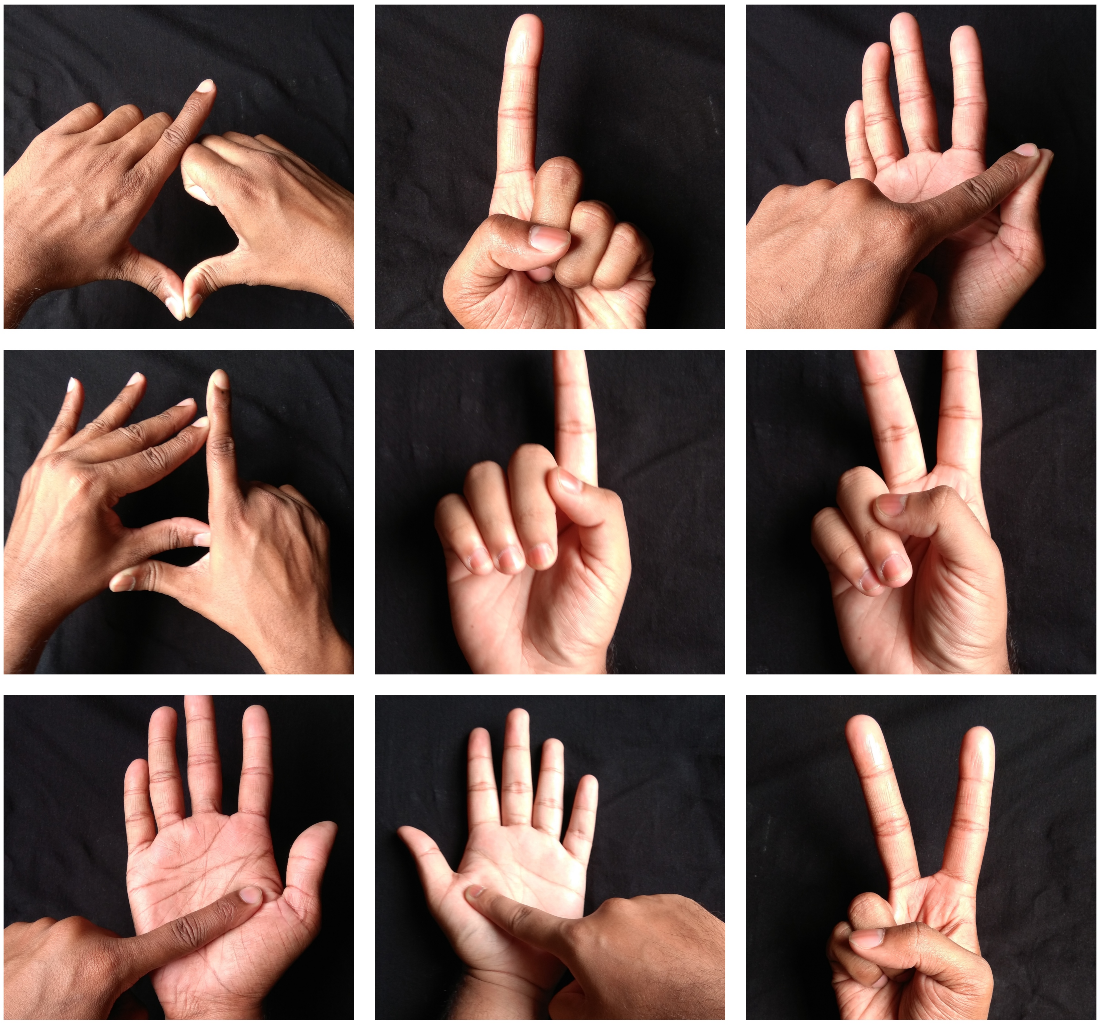

Figure 3.7: Some Bengali hand signs from our extended dataset for testing

The VGG19 can classify close to 1000 objects such as keyboards, mice, pencils and many animals as well [24].

3.7 Inception-ResNet-V2

Inception Resnet v2 is the second sub-version of the hybrid of inception and resnet. Inception-Resnet has two subversions called simply v1 and v2. Both sub-versions have similarities as well as differences. The v1 has a computational cost that is same as inception v3 while the v2 subversion cost is more similar to inception v4. Due to being a hybrid of two other models the stems of v1 and v2 are different. This difference of the stems can be observed when comparing it to inception v1 [27].

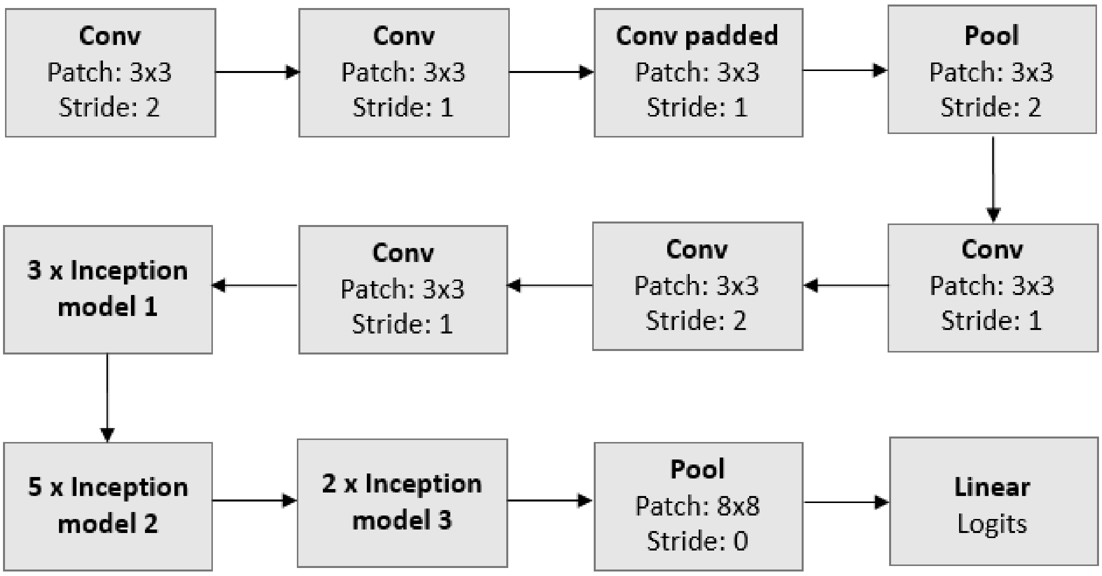

Figure 3.8: The basic architecture of Inception-V3

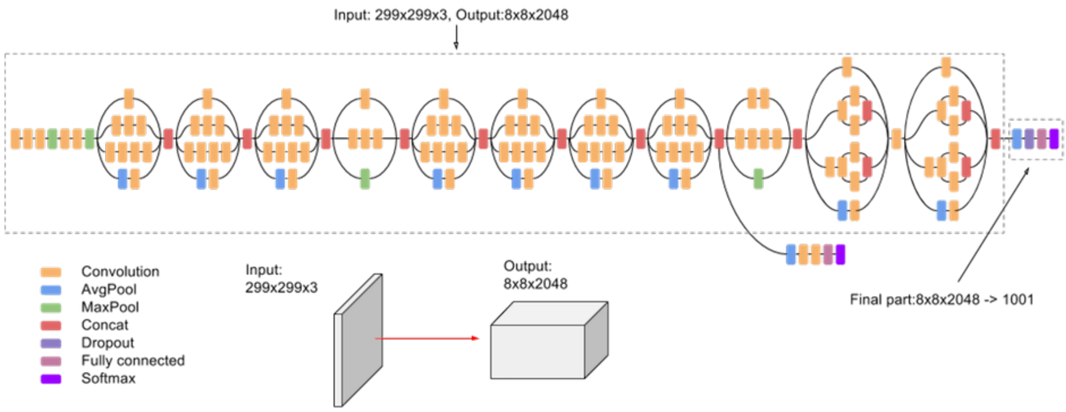

Figure 3.9: Inception-V3 model

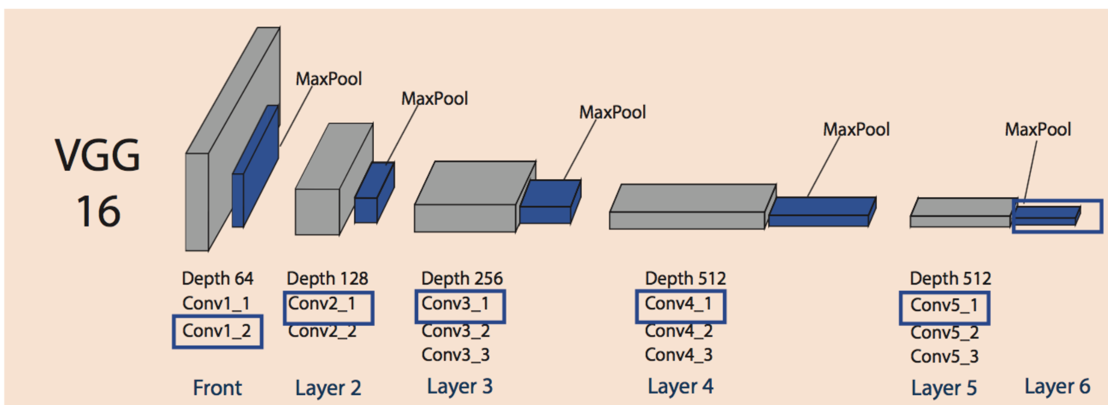

Figure 3.10: VGG16 Model Architecture

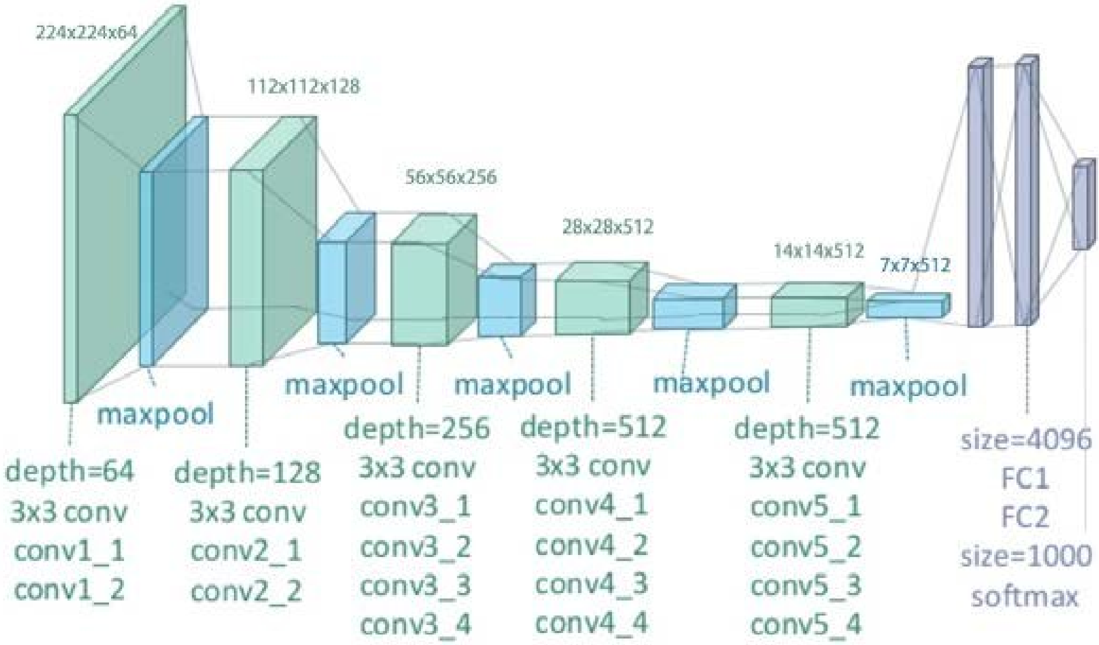

Figure 3.11: VGG19 Model Architecture

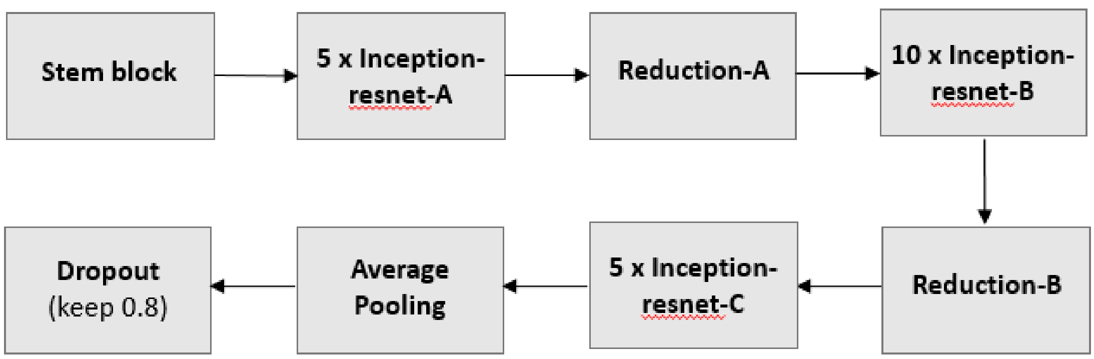

Figure 3.12: Inception-ResNet-V2 Basic Architecture

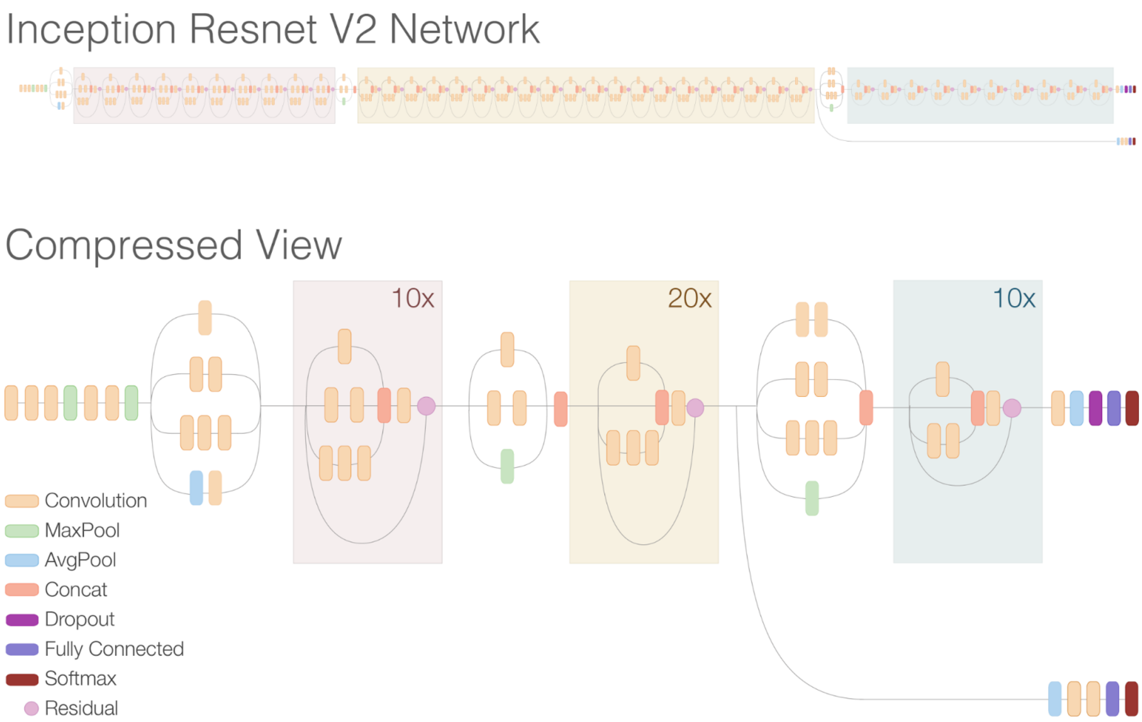

Figure 3.13: Inception-ResNet-V2 Full Architecture

Chapter 4

Experimental Result and Analysis

4.1 Test and training sets

Our dataset consists of 45 classes that include Bengali alphabets and numbers, which from the hand signs of 3 people. These images were also further augmented using scripts. This gives us 1000 images for each class in training, 200 images for testing and another 200 images for validation. So, in total we have 1400*45 = 63000 images.

4.2 Results

We have trained the vgg16, vgg19, inception-resnet-v2 and inception v3 model with the training images without any prior weights. We trained the model layer by layer without any transfer learning as we didn't find any relevant weights for data. To check the training and validation accuracy we have generated the model accuracy between the train and validation graph. Our model will improve more if we increase the epoch number. We have generated these graphs to observe how our models perform on the validation set. And we have generated the model loss between train and validation which shows the comparison between training loss and validation loss.

In the Fig 4.1, the peak training accuracy was about 3.3% while the peak validation accuracy reached 3.1%. There was a lot of fluctuation even when the epoch amount was increased. The mean training and validation accuracy being 2.4% and 2.25%.

In the Fig 4.2, the minimum training loss was about 3.78 while the minimum validation loss reached 3.81. There was an average amount of fluctuation even when the epoch amount was increased which resulted in a mean of 3.79 and 3.82 training and validation loss.

In the Fig 4.3, the peak training accuracy was about 3.4% while the peak validation accuracy reached 3.2%. There was a lot of fluctuation even when the epoch amount was increased.

In the Fig 4.4, the minimum training loss was about 15.56% while the minimum validation loss reached 15.60%. There was a lot of fluctuation even when the epoch amount was increased.

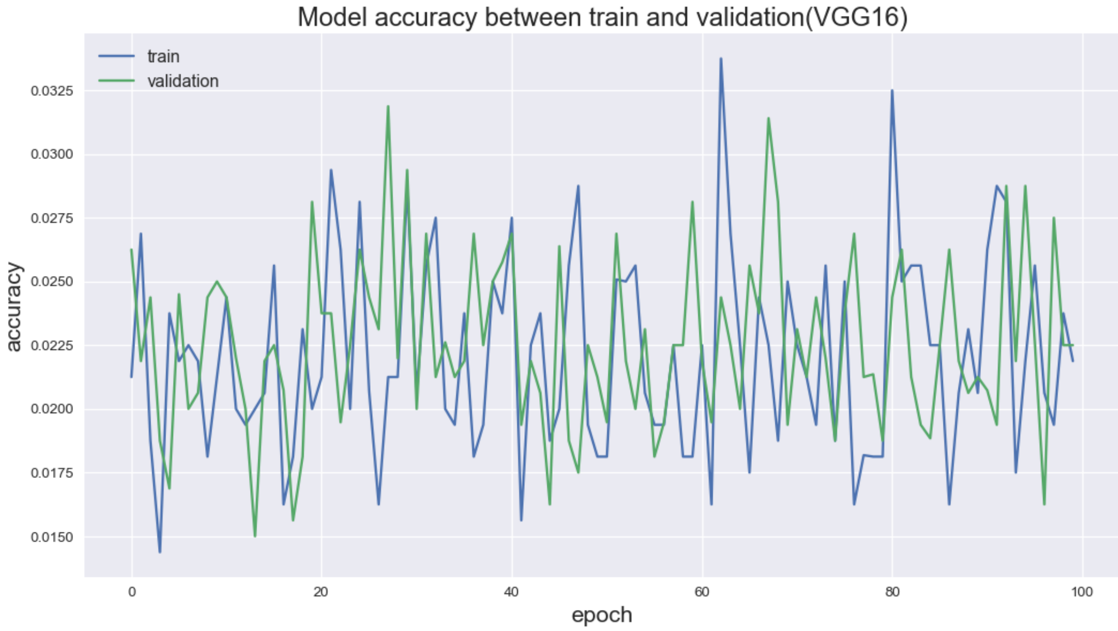

Figure 4.1: Model accuracy between train and validation set (VGG16)

Figure 4.2: Model loss between train and validation set (VGG16)

In the Fig 4.5, the peak training accuracy was about 100% while the peak validation accuracy reached 99.5%. There was a lot of fluctuation even when the epoch amount was increased for validation but not for training.

In the Fig 4.6, the minimum training loss was negligible while the minimum validation loss reached 0.0177%. There was a lot of fluctuation even when the epoch amount was increased for validation but not for training.

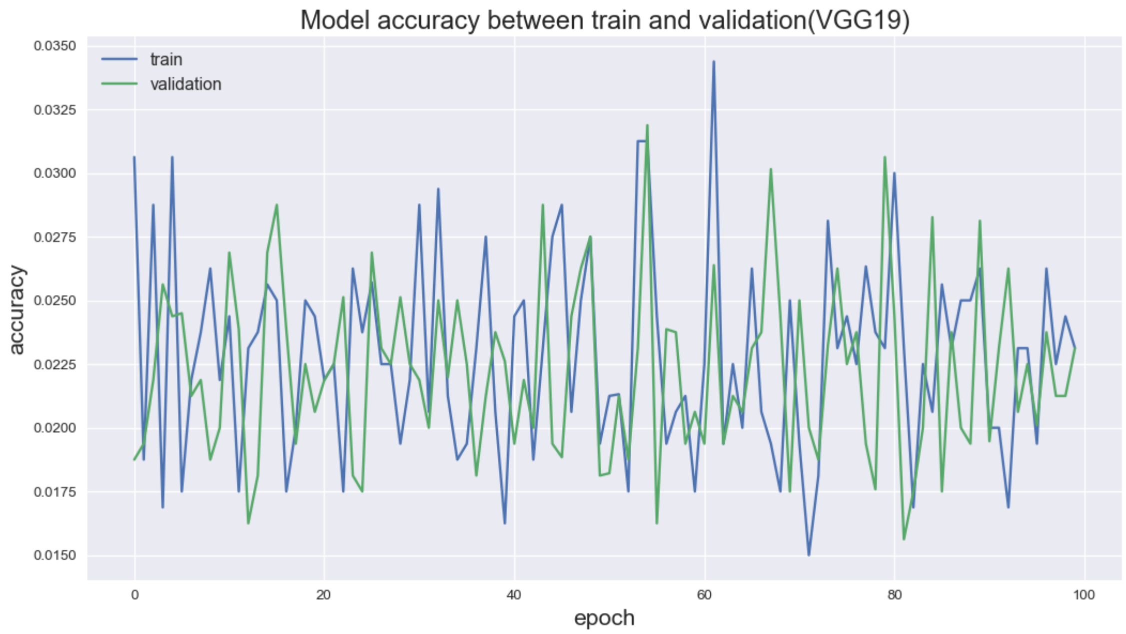

Figure 4.3: Model accuracy between train and validation set (VGG19)

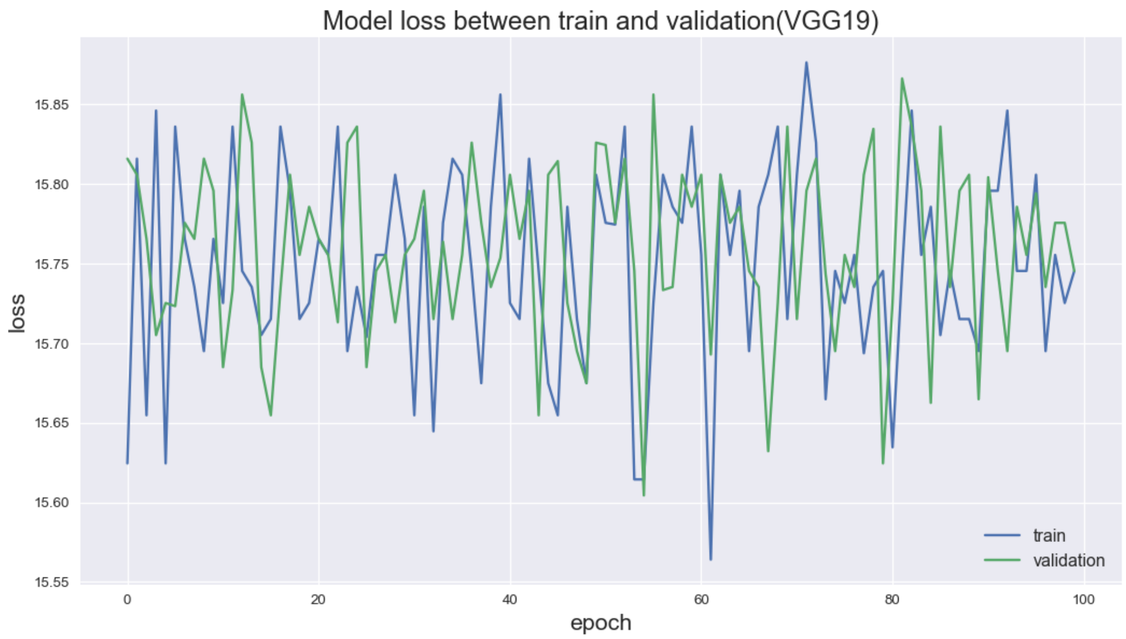

Figure 4.4: Model loss between train and validation set (VGG19)

In the Fig 4.7, the peak training accuracy was about 100% while the peak validation accuracy also reached 100%. There was a lot of fluctuation even when the epoch amount was increased.

In the Fig 4.8, the minimum training loss was negligible while the minimum validation loss reached 0.00012%, which can be taken as negligible. There was a lot of fluctuation even when the epoch amount was increased.

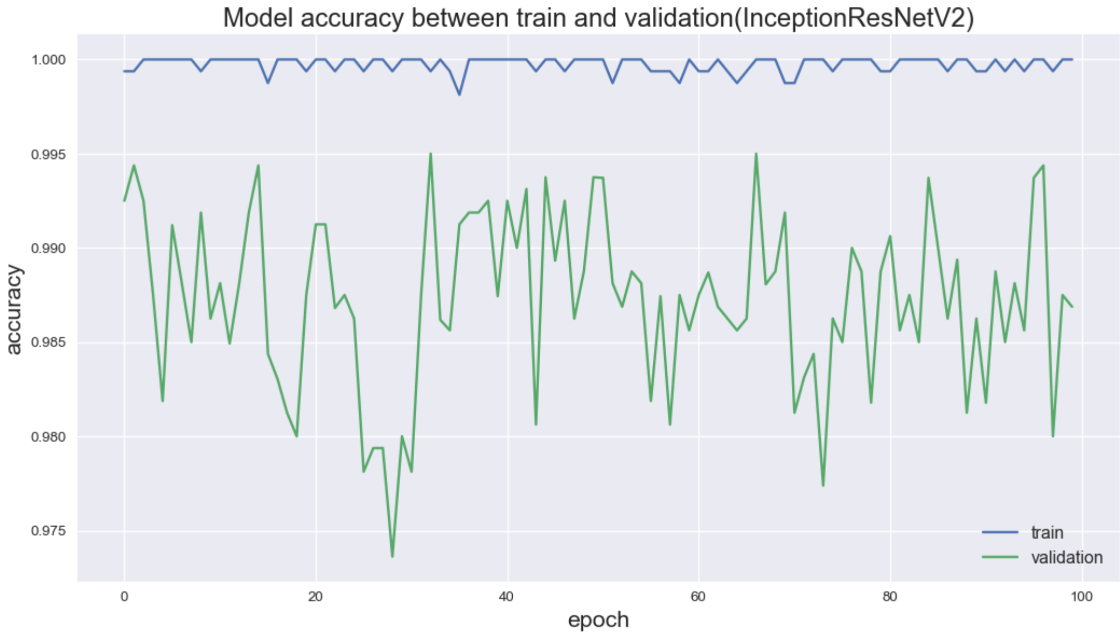

Figure 4.5: Model accuracy between train and validation set (InceptionResNetV2)

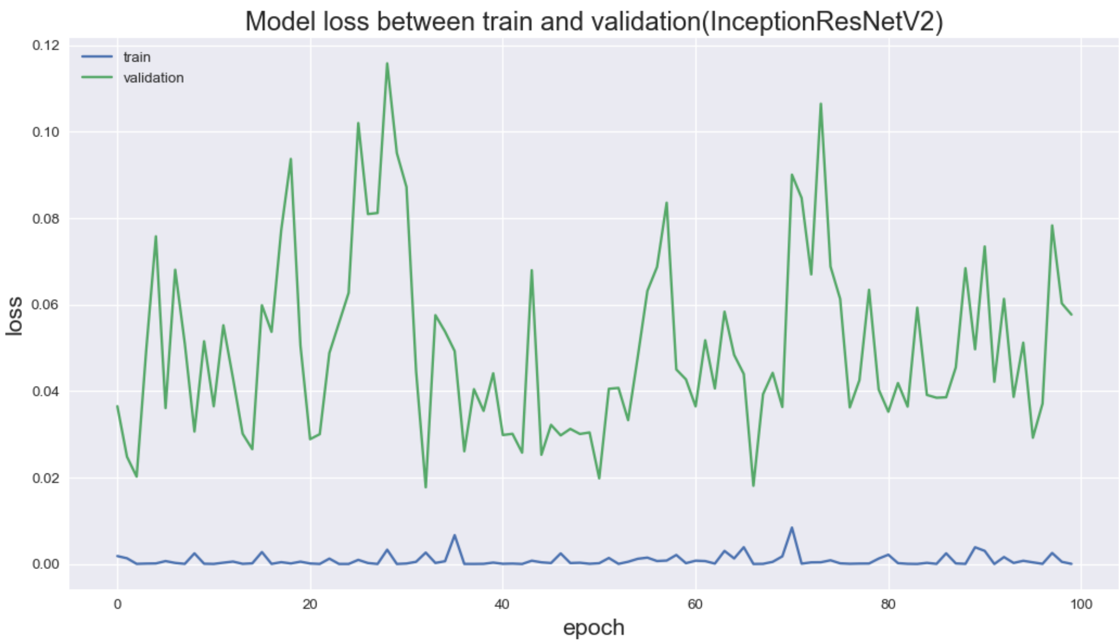

Figure 4.6: Model loss between train and validation set (InceptionResNetV2)

Figure 4.7: Model accuracy between train and validation set (Inception_V3)

Figure 4.8: Model loss between train and validation set (Inception_V3)

4.3 Comparative Analysis

The tables 4.1 and 4.2 refer to all the previous related research works that have been done in the past and similar to our project. The methodologies they have used, the types of dataset they have used in training and testing as well as their average accuracy from their proposed models. There are upsides and downsides to the approaches they have all used.

When we compare the methods of all these projects, they have many similarities with others, mainly the ones that use neural networks. Also, the Support Vector Machine method is also very common among these projects. If we were to discuss about the datasets, most of them used images while one of them used video footage. The amount of images among them range from 260-2400 images of characters. But our dataset contains the most, at 63000 images. Our model also contains more classes, that is it contains more Bengali alphabets and numbers at 45, while the models of others contain about 38 Bengali Hand Sign Language alphabets. But, none of these projects have the complete dataset of all Bengali alphabets.

If we were to compare the methods with ours, for our model the pre processing required per image takes comparatively less effort than the other models. As we are using the Inception v3 model, our feature extraction is carried out within the model, so it takes comparatively less processing power than other approaches. The feature generalization of our model is also better because we are using deep learning. The overall accuracy is also comparatively much better than most methods as we have relatively more classes.

Most of the other proposed models involved the changing of images to grayscale and apply some sort of skin detection algorithms to detect the shape of the hands. However, in our CNN model we are feeding the images with all the 3 color channels, that is we are feeding RGB images directly and not undergoing any grayscale conversion and not running any skin detection algorithm separately. The model extracts features from the images layer by layer, where simple features like edges are detected in the initial layers and more complex features are extracted the deeper we go through the layers.

Authors	Year	Paper Title	Dataset	Methods	Accuracy
S. T. Ahmed et al. [20]	2016	Bangladeshi Sign Language Recognition using Fingertip Position	518 images of 37 BdSL alphabets	Artificial neural network, Canny edge detection, Back propagation	99%
A. R. Chowdhury et al. [25]	2017	Bengali Sign Language to Text Conversion using Artificial Neural Network and Support Vector Machine	Around 800 iterations of 38 BdSL alphabets	Artificial neural network, Support vector machine, Convex hull method	84.11%
M. A. Uddin et al. [23]	2016	Hand Sign Language Recognition for Bangla Alphabet using Support Vector Machine	2400 images	Gabor filter, Kernel PCA, Morphological operation, SVM	99.5%
M. A. Rahaman et al. [15]	2014	Real-Time Computer Vision-Based Bengali Sign Language Recognition	3600 images of 6 Bengali vowels and 30 Consonants	KNN Classifier	96.46%
B. C. Karmokar et al. [11]	2012	Bangladeshi Sign Language Recognition Employing Neural Network Ensemble	235 samples of 47 BdSL signs	NNE, NCL	93%
K. Deb et al. [12]	2012	Two-Handed Sign Language Recognition for Bangla Character Using Normalized Cross Correlation	80 320x240 pixel images	RGB color model, Color segmentation, Template matching	96%
Our proposed model	2019	Bengali Hand Sign Language Recognition Using Convolutional Neural Networks	63000 images of 45 hand signs	CNN, Inception v3	99.89%

Table 4.1: Comparative Analysis

Authors	Year	Paper Title	Dataset	Methods	Accuracy
G. A. Rao et al. [29]	2018	Deep convolutional neural networks for sign language recognition	200 signs of 5 persons in different viewing angles and background environments	CNN	92.88%
Geetha M and M. R. Kaimal [28]	2017	A 3D stroke based representation of sign language signs using key maximum curvature points and 3D chain codes	711 videos from Purdue and Boston with 80 and 77 words from Purdue and Boston respectively	Key Maximum Curvature Points, Stroke Subsegment Vectors, Equivolumetric Partition	85.4%, 82.4%
M. Elbadawy et al. [26]	2017	Arabic sign language recognition with 3D convolutional neural networks	25 gestures from Arabic sign language dictionary	3D Convolutional Neural Network (CNN)	98%
Ravi et al. [7]	2016	Video Based Indian Sign Language Recognition Using Block Zig-Zag DCT Features and Mahalanobis Distance Classifier	20 isolated words are from Indian Sign Language (ISL) are trained and tested	Local cosine feature (LCF), Mahalanobis distance classifier	90.44%
D. Naglot et al. [21]	2016	Real Time Sign Language Recognition using the Leap Motion Controller	520 samples of 26 ASL alphabets	Multi-layer perceptron, Back propagation, Neural network	96.15%
S. N. Sawant et al. [16]	2014	Real Time Sign Language Recognition using PCA	260 images of 26 ISL hand signs	PCA, Morphological filtering, Otsu algorithm	Undefined
Our proposed model	2019	Bengali Hand Sign Language Recognition Using Convolutional Neural Networks	63000 images of 45 hand signs	CNN, Inception v3	99.89%

Table 4.2: Comparative Analysis (continued)

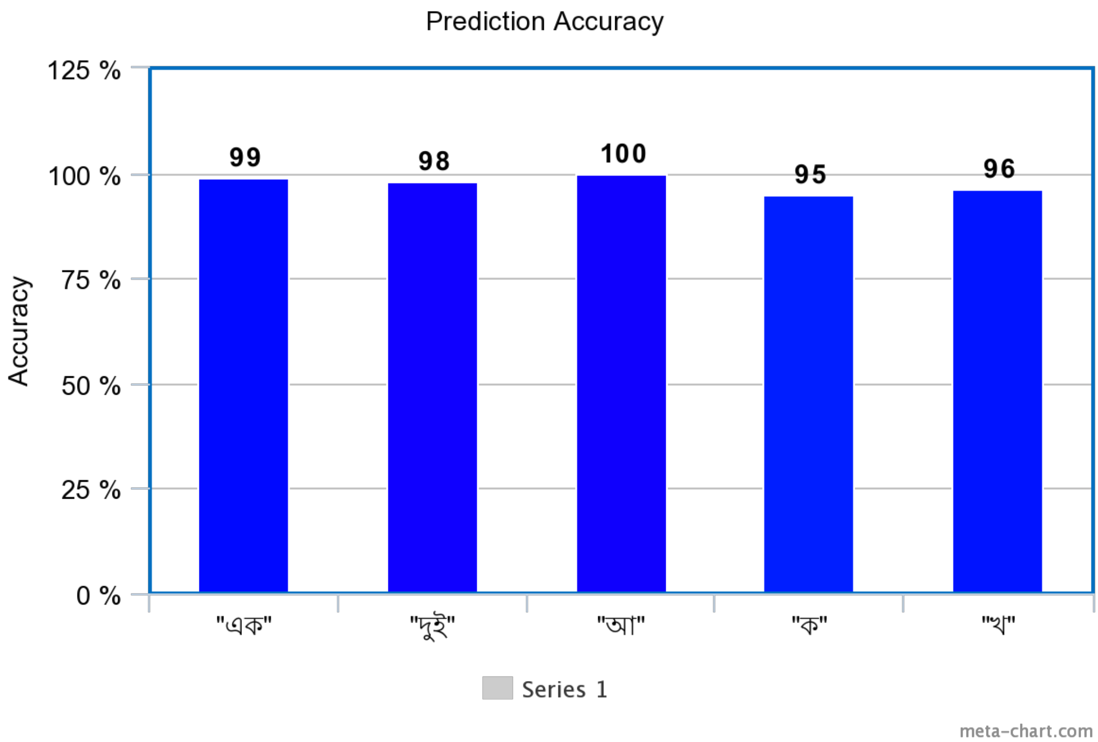

Figure 4.9: Prediction accuracy of some Bengali hand sign characters

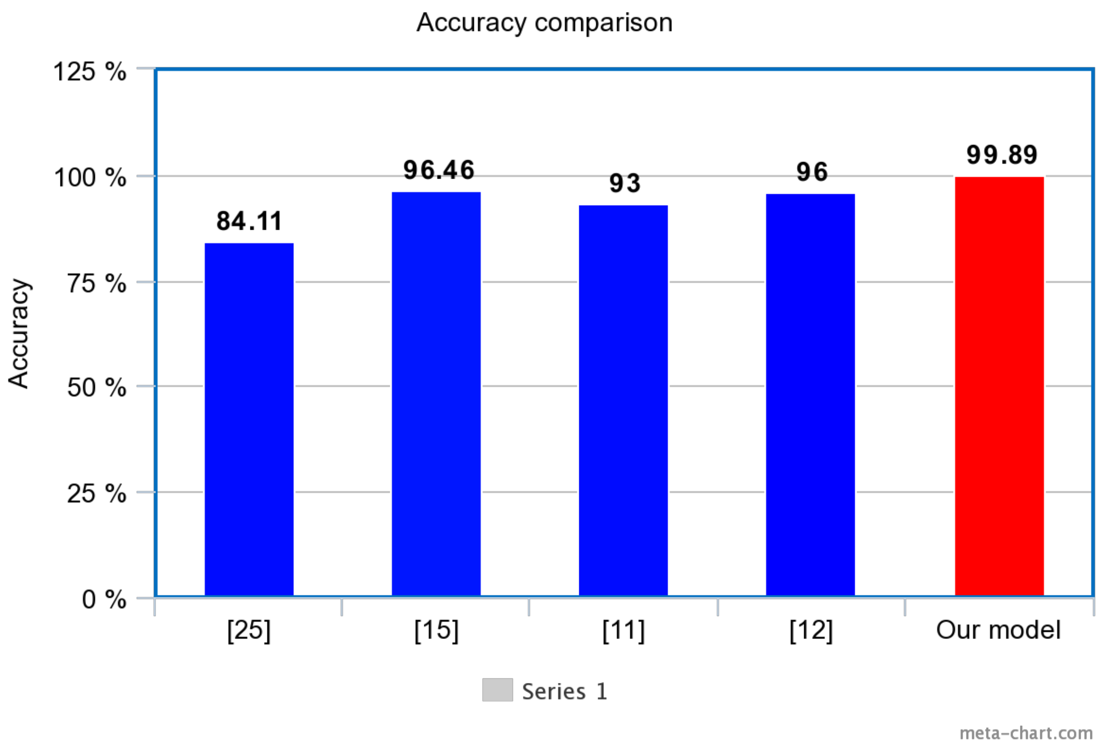

Figure 4.10: Bengali hand sign recognition accuracy comparison of other models and ours

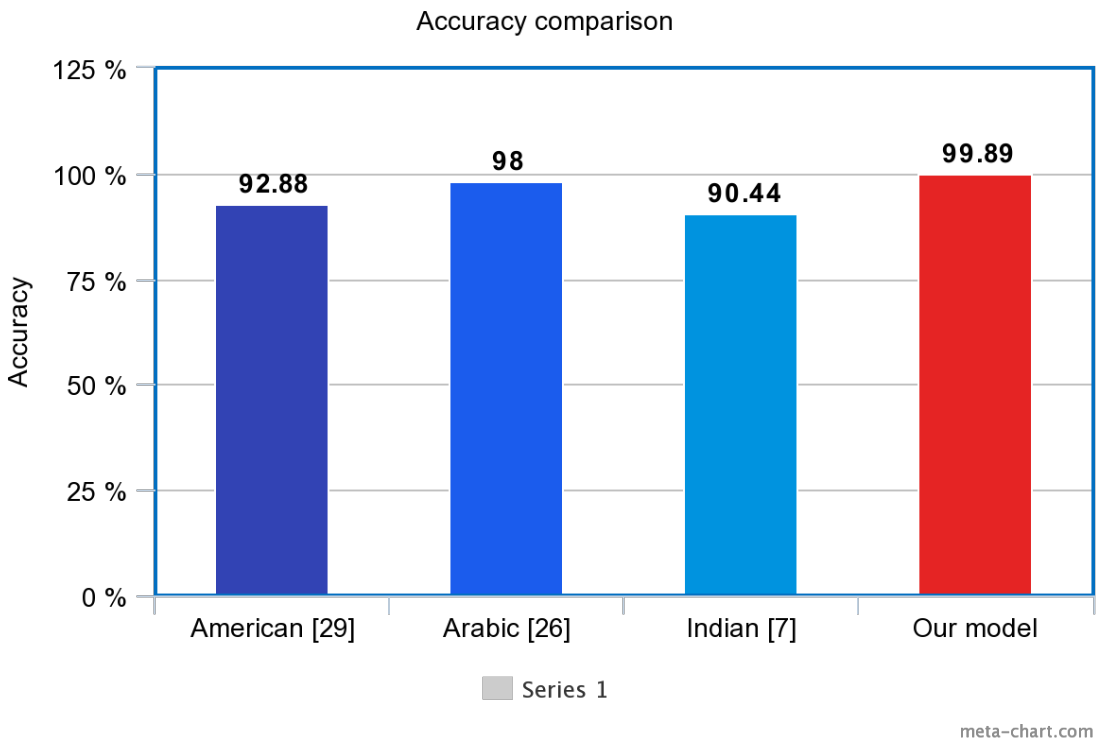

Figure 4.11: Hand sign recognition accuracy comparison between other languages and ours

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

To conclude, the research that we have conducted using the inception_v3 model gave us considerably solid results that can ensure the credibility of our work. The mean validation accuracy that we received is 99.89% and the mean validation loss is 0.0042 which proves that this model is ideal for such recognition purposes.

The inclusion of deaf and mute people into all aspects of our society is very important for us as a nation to further develop and improve and dispel all boundaries and obstacles that hinder our advancement. Our project would serve to fulfill that exact purpose to remove all obstacles, mainly the hindrance of the communication gap between normal people and deaf people. As a result, these people will no longer be wrongly judged and be viewed otherwise negatively in society. They will no longer be left out and will always be treated as equals.

5.2 Future Scope

Our current model takes about one minute to recognize a hand sign from video footage, so it is not in real time. Our plan is to make this process as seamless and as fast as possible and in real time. The dataset that we have used for training, testing and validation does not contain all Bengali alphabets. We intend to complete our dataset by including all alphabets. Gesture recognition has also not been included in our study and neither has been the recognition of words and complete sentences, so we also look forward to including these into our system. After including all of these methods, we plan on integrating our system into a portable device, mainly smartphones and run our system in the form of applications. This way it will be easily accessible to all people and thus help us in introducing such a technology to our society.

Bibliography

[1] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis", *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[2] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function", *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[3] C. B. Barber, D. P. Dobkin, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls", *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[4] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis", in *International conference on artificial neural networks*, Springer, 1997, pp. 583–588.

[5] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers", *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[6] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication", *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 9, pp. 1485–1490, 2005.

[7] S. Ravi, M. Suman, and P. Kishore, "Video based indian sign language recognition using block zig-zag dct features and mahalanobis distance classifier", 2006.

[8] P. Vogt, K. H. Riitters, C. Estreguil, J. Kozak, T. G. Wade, and J. D. Wickham, "Mapping spatial patterns with morphological image processing", *Landscape ecology*, vol. 22, no. 2, pp. 171–177, 2007.

[9] G. Murthy and R. Jadon, "Hand gesture recognition using neural networks", in *2010 IEEE 2nd International Advance Computing Conference (IACC)*, IEEE, 2010, pp. 134–138.

[10] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition", in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, IEEE, 2011, pp. 342–347.

[11] B. C. Karmokar, K. M. R. Alam, and M. K. Siddiquee, "Bangladeshi sign language recognition employing neural network ensemble", *International journal of computer applications*, vol. 58, no. 16, pp. 43–46, 2012.

[12] D. Kaushik Deb, M. I. Khan, H. P. Mony, and S. Chowdhury, “Two-handed sign language recognition for bangla character using normalized cross correlation”, *Global Journal of Computer Science and Technology*, 2012.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] H. J. Vala and A. Baxi, “A review on otsu image segmentation algorithm”, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 2, pp. 387–389, 2013.

[15] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman, “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.

[16] S. N. Sawant and M. Kumbhar, “Real time sign language recognition using pca”, in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, IEEE, 2014, pp. 1412–1415.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.

[18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *arXiv preprint arXiv:1502.03167*, 2015.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[20] S. T. Ahmed and M. Akhand, “Bangladeshi sign language recognition using fingertip position”, in *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, IEEE, 2016, pp. 1–5.

[21] D. Naglot and M. Kulkarni, “Real time sign language recognition using the leap motion controller”, in *2016 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, vol. 3, 2016, pp. 1–5.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[23] M. A. Uddin and S. A. Chowdhury, “Hand sign language recognition for bangla alphabet using support vector machine”, in *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, IEEE, 2016, pp. 1–4.

[24] T. Carvalho, E. R. De Rezende, M. T. Alves, F. K. Balieiro, and R. B. Sovat, “Exposing computer generated images by eye’s region classification via transfer learning of vgg19 cnn”, in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017, pp. 866–870.

[25] A. R. Chowdhury, A. Biswas, S. F. Hasan, T. M. Rahman, and J. Uddin, "Bengali sign language to text conversion using artificial neural network and support vector machine", in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, IEEE, 2017, pp. 1–4.

[26] M. ElBadawy, A. Elons, H. A. Shedeed, and M. Tolba, "Arabic sign language recognition with 3d convolutional neural networks", in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE, 2017, pp. 66–71.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning", in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] M. Geetha and M. Kaimal, "A 3d stroke based representation of sign language signs using key maximum curvature points and 3d chain codes", *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7097–7130, 2018.

[29] G. A. Rao, K. Syamala, P. Kishore, and A. Sastry, "Deep convolutional neural networks for sign language recognition", in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, IEEE, 2018, pp. 194–197.