Sign Language Translation

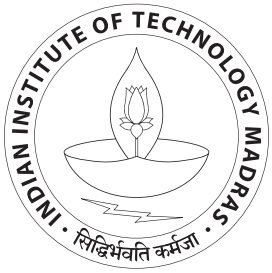
A Project Report

submitted by

ADVAITH SRIDHAR

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.
May 2019

THESIS CERTIFICATE

This is to certify that the thesis entitled **Sign Language Translation**, submitted by **Advaith Sridhar** (**EE15B004**), to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelors of Technology** is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Avhishek Chatterjee
Project Guide
Asst. Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to express my deepest respect and sincere gratitude to my guide Dr. Pratyush Kumar for his continuous support and invaluable guidance throughout the entire project. I would also like to thank Dr. Avhishek Chatterjee for accepting me for this project.

Apart from them, my heartiest thanks to the Ability Foundation and St. Louis's School for the Deaf, for generously agreeing to commit their time and resources to help us build an Indian Sign Language Dataset.

Equally important is the support I received from my friends Rohith, Shubham and Aravinth who helped me throughout the course of this project.

Finally, I would like to thank my friends, family and IIT Madras for their constant support and encouragement, without which I would not be where I am today.

ABSTRACT

KEYWORDS: Indian Sign Language, Gesture Recognition, OpenPose, Clustering, Limb Action States

Sign Language is used by the deaf and dumb community to communicate with each other. There are several types of Sign Language, each having it's own unique gestures and properties. Recently, several attempts have been made to automatically translate sign language, using datasets available for American, British and Chinese Sign Language. Methods involving CNNs and RNNs for pose estimation, nearest neighbour estimation and Markov Random Fields have been tried for the same. There is a dearth of datasets for Indian Sign Language, which has prevented progress in any translation attempt for the same. In this work, we create the first comprehensive Indian Sign Language dataset, and then provide an approach which uses a mix of techniques to classify various signs. We incorporate a pre-trained Deep Learning model to do pose detection and extract key skeletal points from the input data. Next, we introduce a novel approach feature extraction by defining limbs from the pose data and observing their properties. Action states are identified for each limb by clustering, and temporal and spatial connections between various limbs are analyzed and used to aid classification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ISL	Indian Sign Language
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
kNN	k-Nearest Neighbours
tSNE	t-Distributed Stochastic Neighbor Embedding
PAF	Part Affinity Field
ANN	Artificial Neural Network

CHAPTER 1

INTRODUCTION

1.1 Indian Sign Language

There are 1.1 million deaf-and-dumb people in India. 98% of these people are illiterate, making sign language their only method of communication. This trend is not expected to change anytime soon, as only 2% of deaf children attend school.

Just like spoken languages, sign language has many different variations. Sign language from different regions can vary significantly, not just in the actions they use for various signs, but also the body parts involved in signing. For example, American Sign Language uses only a single hand to sign, whereas British Sign language (which is the inspiration behind ISL) uses both hands. Within India too, a multitude of Sign Languages exist. The Bombay-Madras Sign Language, Calcutta Sign Language, Delhi Sign Language and Bangalore Sign Language all use different actions and body parts for communication.

The Ministry of Social Justice and Empowerment, Government of India, launched the first ISL dictionary last year, defining an official Indian Sign Language in the process. For the purpose of this project, any mention of ISL refers to this official dictionary, and all signs recorded for the creation of an ISL dataset follow the signs displayed in this dictionary.

1.2 Project Aim

The aim of this project was to build an automatic ISL translator- i.e. given an input video of someone signing a word, the model should be able to predict which word is being signed. Such a model has several use-cases, and can be deployed either through an app or website for people to use. It has high relevance in the lives of not only the deaf-dumb, but also anyone who interacts with them.

One of the major challenges that we faced was that there are no ISL Datasets available online. Though datasets for Chinese, British, German, American and other country datasets are available, there exist absolutely none for Indian Sign Language. Finding NGOs / universities to collaborate with to generate data was also an incredible hassle, and we are incredibly grateful to the Ability Foundation and St. Louis for their cooperation in this regard.

1.3 Report Flow

This report first introduces the previous work done in this field (Chapter 2), then moves on to describe the dataset generated for this project (Chapter 4). Next, the methodology used is described in detail in Chapter 5. In case some background material is required to understand the techniques used in Chapter 5, it can be found in Chapter 3, Background. Finally, we summarize our efforts and describe the scope for future work in Chapter 6.

CHAPTER 2

PREVIOUS WORK

There have been a lot of sign language translation attempts with custom made datasets. Most of these datasets are heavily limited in scope (only images, videos in uniform background such that hands can be easily segmented etc.), which makes the model unusable in a general context. Some of the work observed are:

- Recognition of sign language in live video by Singha and Das [2013] works only on detecting hands from uniform dark background by converting the image to the HSV space. It then uses histograms to detect similarity between images and selects unique images.Finally, images are passed through a classifier to predict signs.
- Nearest Neighbour classification of ISL using kinect cameras by ANSARI and HARIT [2016] uses kNNs to classify 140 different signs. The paper uses a dataset with only images, and makes a lot of assumptions about the data available.
- Selfie video based ISL recognition system by Rao and Kishore [2018] again depends on contrast between the hands and the background, which it detects using an edge detector. After this, some feature extraction is done and kNN is used for classification.
- Low latency human action recognition by Cai *et al.* [2016] detects skeletal key points from a human gesture. It uses these to define limbs or connections between adjacent points on the skeleton. Velocity and acceleration of these points are calculated as well, and a Markov Random Field is used to label test data limbs into states. Classification is done by observing histograms of this test data.
- Segment, track, extract, recognize and convert sign language by Kis classifies 351 symbols using an ANN. The model relies on a subject wearing a tshirt of the same colour as the background wall, which should contrast with his/her hands and face.

CHAPTER 3

BACKGROUND

3.1 OpenPose

OpenPose by Cao *et al.* [2018] is the first real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images and videos. OpenPose has the following features:

- Input: Image, video, webcam, Flir/Point Grey and IP camera.
- Output: Basic image + keypoint display/saving (PNG, JPG), keypoint saving (JSON), and/or keypoints as array class.

The output pose points are 2D and consist of the following:

- 25-keypoint body/foot keypoint estimation.
- 2x21-keypoint hand keypoint estimation.
- 70-keypoint face keypoint estimation.

OpenPose uses a bottom-up approach to pose detection. It uses a CNN-RNN model to calculate Confidence score estimates for each body point, as well as Part Affinity Fields(PAFs). The network uses the first 10 layers of VGG-19 to generate a set of feature maps, which it uses as input to the first stage of its network. The first stage outputs a set of PAFs and confidence scores, which it concatenates with the feature Map before running it through the network again.

For the purpose of this project, we use OpenPose as a blackbox in order to generate keypoints, without changing the underlying architecture.

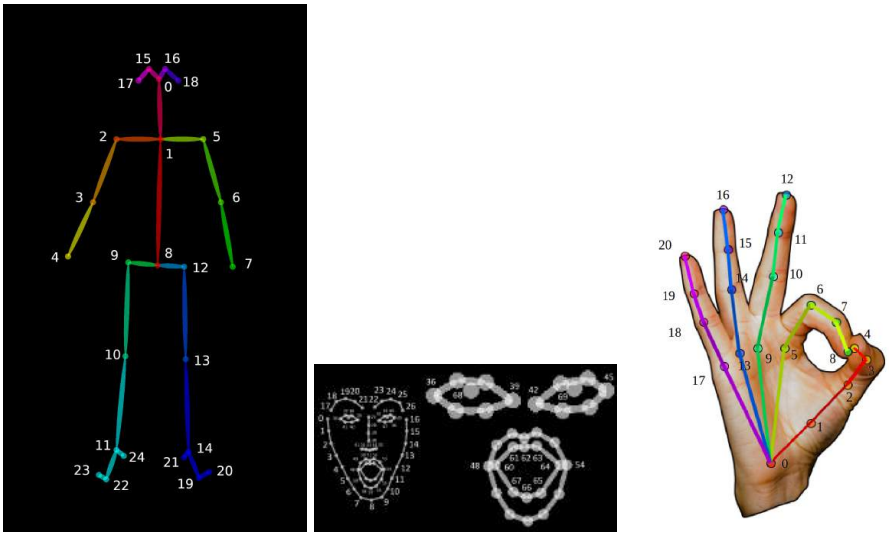


Figure 3.1: Body, Hand and Face Keypoints Provided by OpenPose

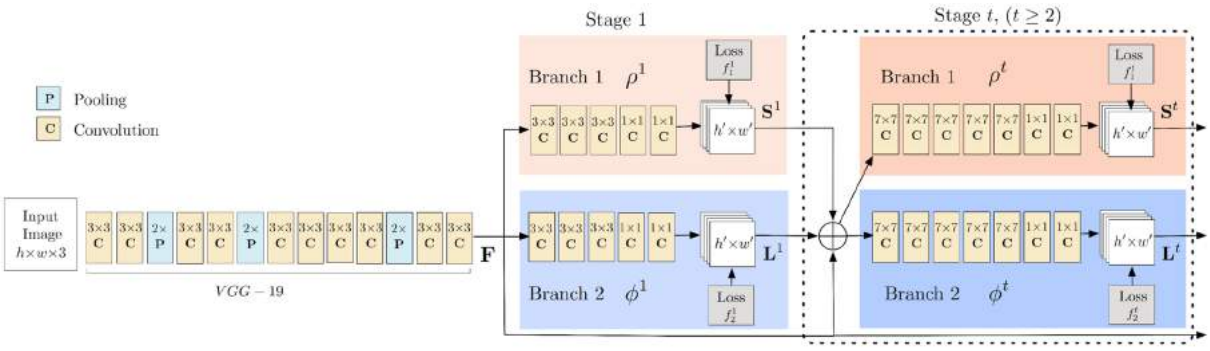


Figure 3.2: The OpenPose architecture

3.2 tSNE

tSNE by van der Maaten and Hinton [2008] is a tool to visualize high dimensional data. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. t-SNE has a cost function that is not convex, i.e. with different initializations we can get different results.

tSNE is a powerful tool to visualise data in 2 or 3 dimensions. In this project, it has been used extensively to visualise human pose points in the 2D space.

3.3 kMeans

The KMeans algorithm clusters data by trying to separate samples in groups which have equal variance. kMeans minimizes the within cluster sum of squares criterion. The K in KMeans is the number of clusters to be used, and is a hyperparameter specified by the user. The algorithm starts with random cluster centroids, and can give different outputs with every run. It has a tendency to get stuck at local minima, and hence should be run multiple times.

3.3.1 Silhouette Score

The Silhouette score is a measure of cluster purity, commonly used when no ground truth labels are present. It takes a value between -1 and +1, where a high value means that the clusters are pure (intra cluster distance is small, and inter-cluster distance is large), while a small value indicates that the clusters are not well

defined. Formally, for each datapoint i, let

$$a(i) = \frac{1}{|C| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

be the average distance between i and all other data points in the same cluster, where $d(i, j)$ is the Euclidian distance between data points i and j in the cluster C_i. We interpret $a(i)$ as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment). We also define

$$b(i) = \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)$$

The Silhouette score is now defined as

$$S = \frac{1}{n} \sum_{i=1}^{n} \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

3.4 Cramer's V Coefficient

Cramer's V is a measure of association between two nominal variables, giving a value between 0 and 1 (inclusive), where 0 stands for independence or no association, while values close to one stand for high degree of association. It is based on Pearson's Chi-Squared statistic.

3.5 Decision Trees

Decision Trees are a non-parametric supervised learning method used for classification. The Decision Tree algorithm predicts a target variable by learning simple

decision rules inferred from the data features. They are especially useful with categorical data, and are hence used in this project to classify based on limb states. They are however prone to overfitting, and must be used carefully.

CHAPTER 4

DATASET

There is currently no publicly available dataset for ISL. Hence, as a part of this project, we attempted to create the first such dataset.To know more about the various existing sign language datasets, refer to Chapter 3. Three types of datasets have been created for different phases of the project. They are:

4.1 Dataset 1: Picture Dataset

The aim of the first dataset was to check if, given pictures of different human poses, the keypoints extracted from OpenPose were sufficient to recognise them as different actions. The dataset contains 120 pictures belonging to 6 different classes, with 20 pictures for each class.
6 signs with differently distinct actions were chosen (based on the recommendations of the Ability Foundation) for this dataset. The 6 signs are:

- Benefit
- Free
- Cash
- Penalty
- Thank You
- Job

The images were shot on an iPhone 6 camera, with resolution 3264(width) x 2448 (height). All 120 pictures were taken in an indoors brightly lit setting, featuring a single person standing against a non-uniform background. The total size of the dataset is 128.1 MB.

Figure 4.1: Images from the dataset.From Left to Right: Line 1-Benefit, Free, Cash. Line 2-Penalty, Thank You, Job

4.2 Dataset 2: Banking Dataset

The aim of this dataset is to record signs used for one specific use case - in this case common words used when a person goes to a bank. The dataset contains 586 videos from 57 classes, with roughly 10 videos per word. The total size of the dataset is 2.5 GB. The list of words and number of videos per word have been

attached in Appendix A.

The words chosen for this dataset were selected by sign language interpreters at the Ability Foundation. All videos are signed by the same person. Each video is between 2-4 seconds in length. Around half of the videos have dimensions of 2160x3840 with a frame rate of 60 fps, while the other half have dimensions 1920x1080 with a frame rate of 30fps.The dataset was shot using 2 handheld smartphone cameras kept about 4 feet away from the subject. The videos were recorded from different angles, in bright lighting conditions with a non-uniform background.

Figure 4.2: Video frames from the dataset. The words from Left to Right: Password, Balance, Income

4.3 Dataset 3: General Dataset

This dataset is a general purpose dataset with recordings of the 200 most commonly used words in ISL. The dataset contains around 4600 videos belonging to 264 classes, with roughly 20 videos per class. The total size of the dataset is 54.5 GB.

The list of words and the number of videos per word can be found in appendix B. The words for this dataset were chosen by us and the St. Louis's College for the

Figure 4.3: Recording the dataset

Deaf. 6-7 interpreters were used to record every sign, with each interpreter being the subject of 3-4 videos per sign. Each video is between 1-4 seconds in length. All videos are of dimensions 1920x1080 and have been shot at a frame rate of 25 frames per second. The videos were shot using DSLR cameras mounted on a stand 4-5 ft away from the subject. All videos have been recorded in bright lighting conditions with similar backgrounds.

Figure 4.4: Frames from the general dataset. Video to the left is "Home", and to the right is "Street"

CHAPTER 5

METHODOLOGY

Our methodology can be broken into 2 parts. The first part focused on classifying the Picture Dataset [Section 4.1], and the second part is focused on processing the General Dataset [Section 4.3].

5.1 Image Classification

The first task at hand was to see if skeletal points were enough to distinguish between various human poses. We wanted to test the hypothesis that the skeletal points obtained from OpenPose were sufficient in order to uniquely distinguish between various human poses. Though intuitively they seem so, we decided to check and see if different poses actually occupied significantly different regions in the skeletal point space. If this were not the case, we would have to rethink our pose detection algorithm. We understand that the ability to classify images successfully does not imply the ability to successfully classify videos, however the opposite would hold true - the failure to classify images would imply that the model would fail to classify videos as well.

5.1.1 Step 1: Running OpenPose

We ran each image through OpenPose individually, running with pose and hand flags. Facial keypoints were ommitted as facial expressions were not required for recording the signs in the Picture Dataset. We obtained a list of 67 x,y coordinates per image, which can be seen in figure 5.1:

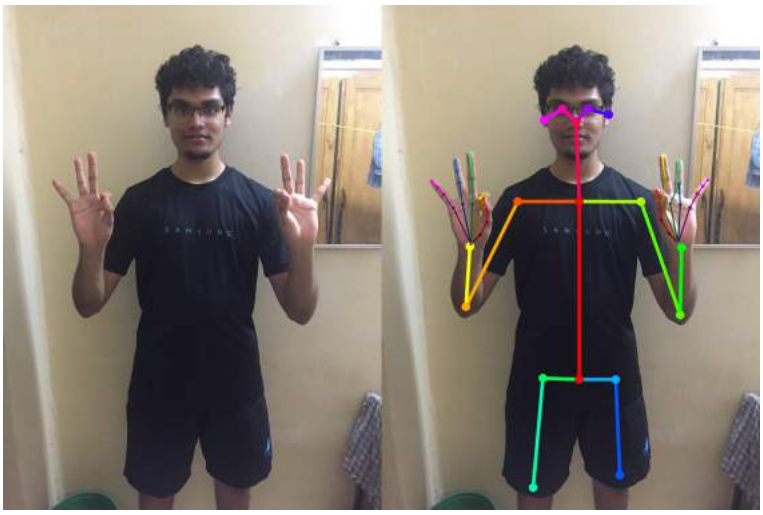


Figure 5.1: Images before and after running OpenPose. The picture depicts the word Free.

5.1.2 Step 2: tSNE

Next, we try to visualise the data. We use tSNE to reduce the feature dimensions from 134 (67 points per picture, x and y coordinates per point) to 2. Upon plotting the 2D data, we observe that different words tend to form unique clusters in the feature-reduced space. Given below is a visualisation of the data:

We observe that all words tend to form unique clusters, though Penalty and Job have some degree of overlap between them. Intuitively too,we see that the pictures for Penalty and Job have a great amount of similarity between them, which possibly explains the overlap in clusters. The words Thank You, Cash and Benefit each form

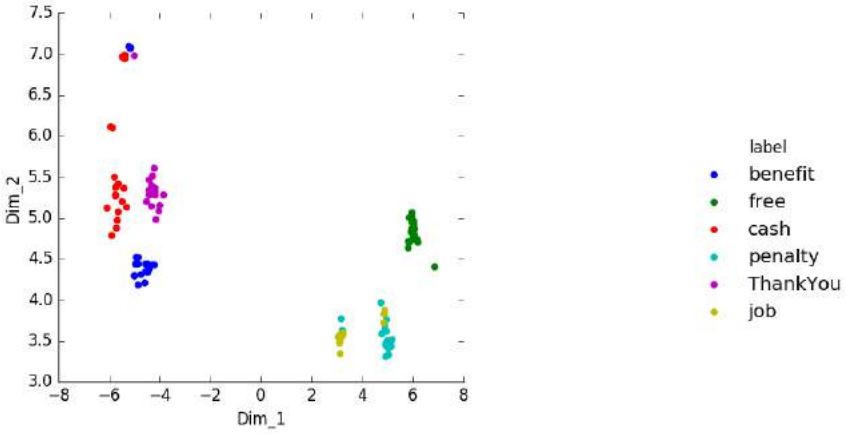


Figure 5.2: tSNE plot of the picture dataset. As it can be observed, different words form different clusters in the dimensionality reduced space

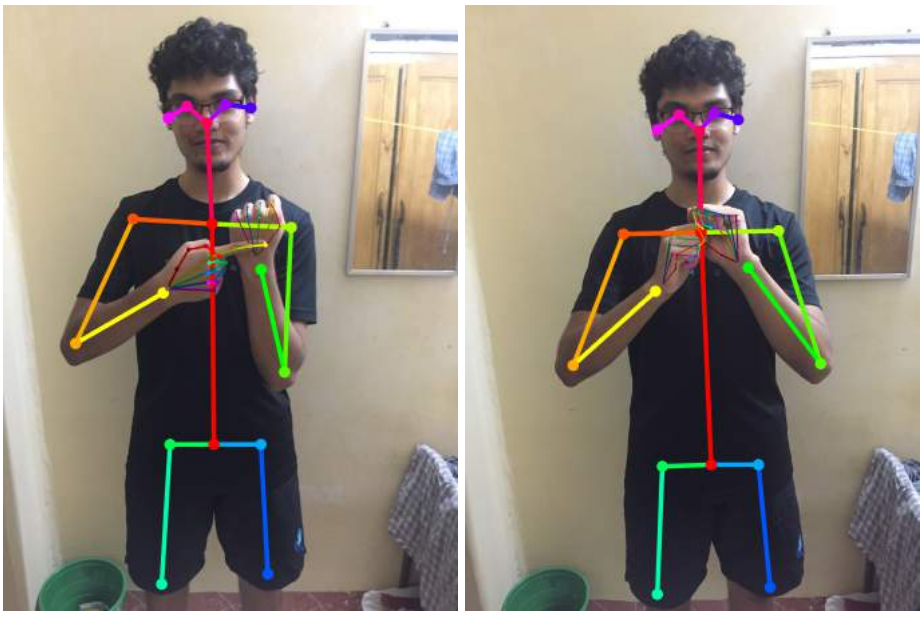


Figure 5.3: Images Penalty (left) and Job (right). We observe that the 2 poses are reasonably similar to each other.

their own clusters, and are close to each other in the feature space. (Fig 5.4) All 3 signs have the left arm in a similar position, with significant differences only in the right arm and fingers. This shows that finger pose points can contribute to significant differences as well. Finally, we see that the Free forms its own unique

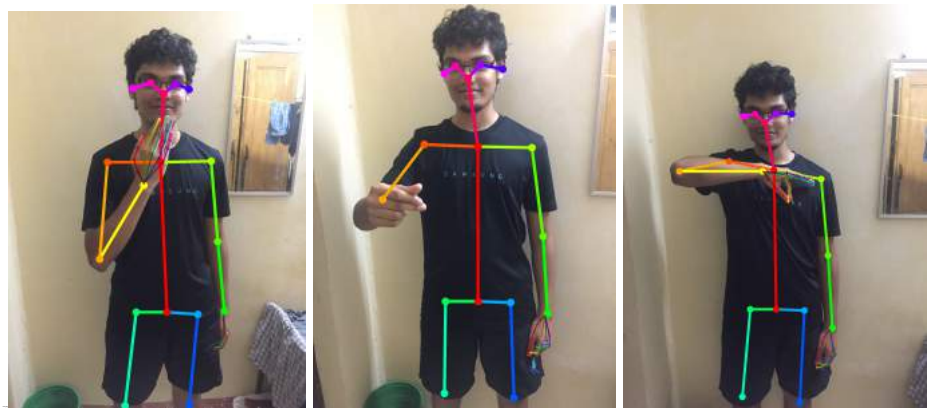


Figure 5.4: Images Thank You(left), Cash(Middle) and Benefit(right).Right arm gestures are the major difference between the 3 images.

cluster in the feature reduced space. Upon observing the image (Fig 5.5), we see that Free's pose is very different from the rest of the words as well. This suggests that drastically different poses form their clusters in the feature space and can be easily separated as well.

5.1.3 Classification

Finally, we train a Decision Tree model on our Pose data. We split the dataset into training and testing datasets, with 90 datapoints (75%) as the training set and 30 datapoints (25%) as the test set. The Gini Impurity Index was used as the criterion for training the model. The Decision Tree hyperparameters were as follows:

- min_samples_split = 2. This is the minimum number of samples required to split an internal node.
- min_samples_leaf = 1. The minimum number of samples required to be a leaf node.

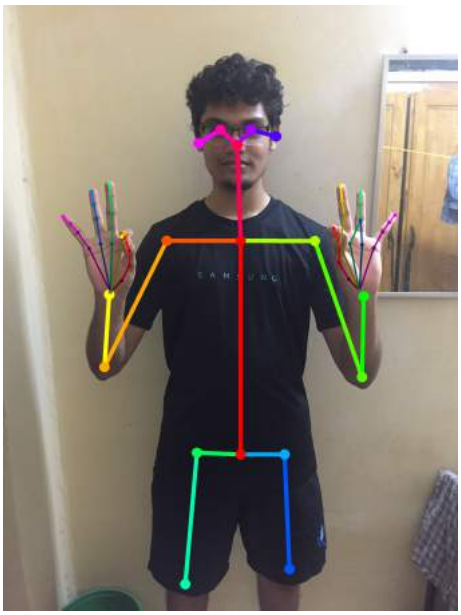


Figure 5.5: The image Free

The following were our results: With results this strong on image classification,

Table 5.1: Model accuracy on original and feature reduced data

Data	Accuracy
Original Data	96.3%
tSNE Data	93.0%

we moved on to our next task, video classification.

5.2 Video Classification Pipeline

For trying video classification, we use the General Dataset [Section 4.3]. Written below is a step-by-step description of the pipeline used for video classification.

5.2.1 Running OpenPose on Videos

First, we run OpenPose on the video,extracting keypoints for every frame. Though we extract all 137 keypoints, we ignore facial keypoints later on. This effectively leaves us with 67 keypoints per frame. Most videos are between 60-90 frames long, and this leaves us with between 4000 and 6000 keypoints per video. Since each keypoint is an (x,y) coordinate, we effectively have between 8000 and 12000 features per video.

Confidence Scores

Apart from (x, y) coordinates for every skeletal point, OpenPose also produces a confidence score for each point. Upon observing the confidence scores for various points in the dataset, it was observed it reported high confidence scores for all facial feature points (between 0.8-1) and relatively low confidence scores for hand keypoints (ranging between 0.1-1 with a mean of 0.45).

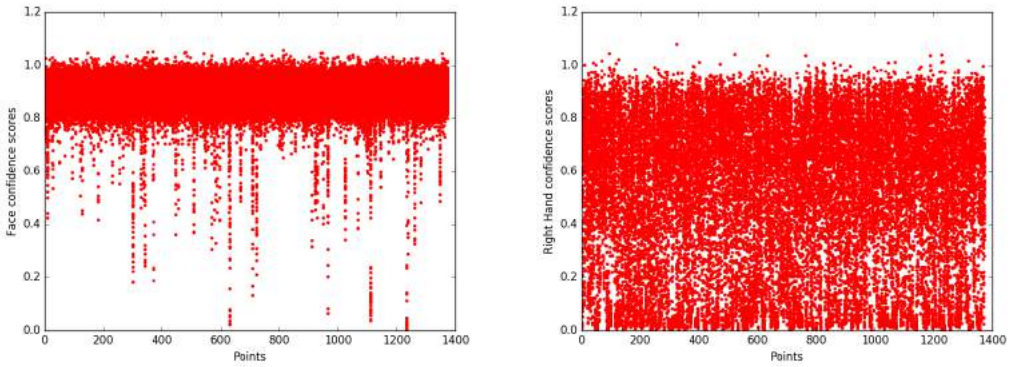


Figure 5.6: Facial points confidence scores (left) and right hand confidence scores (right)

Need for feature extraction

We have 20 videos per class. Splitting this into training and testing datasets, (with 80% used for training and 20% for testing), that leaves us with 16 videos per class available for training.

Given that we have such a low number of training samples (16 per class), and such a large number of features per training sample (8000-12000) it becomes evident that directly applying Machine Learning or Deep Learning algorithms to do classification will not work. Hence, we follow a different pipeline instead, by first defining features from the skeletal points and then defining characteristics of these features. We use these characteristics to cluster similar datapoints and define states, and finally use these states for classification.

5.2.2 Feature Extraction

Next, we extract features from the pose data. We define the concept of limbs, inspired by the Cai *et al.* [2016] paper on skeleton representation.

A limb is defined as a skeleton segment connecting any 2 adjacent points on the human arm. Formally, let $\mathbf{j} = (x, y)$ denote the 2D coordinate of a skeletal point. Let l denote one of the limbs. The Position vector $\mathbf{P}_l$ is defined as:

$$\mathbf{P}_l = \mathbf{j}_{l,end} - \mathbf{j}_{l,start}$$

For our purpose,we consider only limbs found on the arm and the fingers of the human body. For the arms, This leaves us with 4 limbs, 2 on the left arm (connecting points (2,3) and points (3,4)),and 2 on the right (connecting points (5,6) and points (6,7)).

Limbs for fingers are defined similarly. It can be observed from the Figure 5.8 that

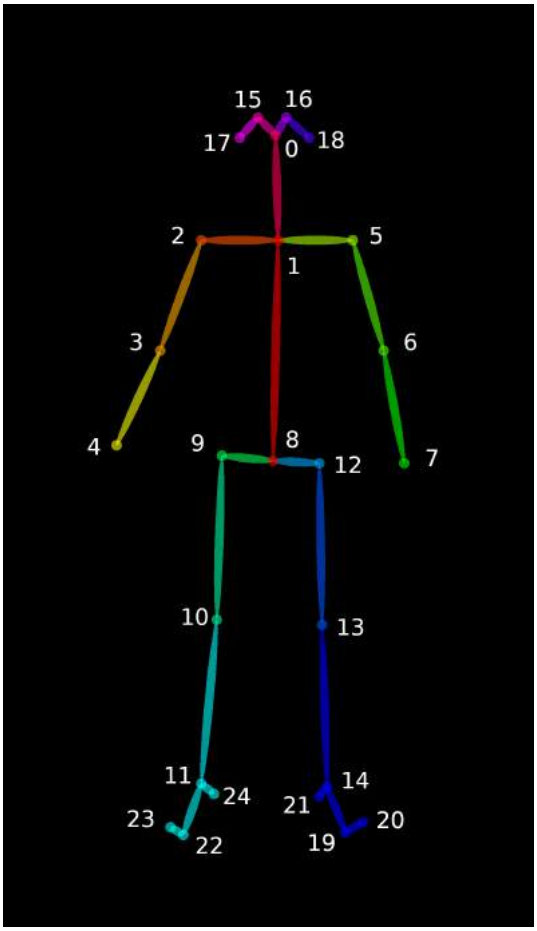


Figure 5.7: Skeletal body points extracted by OpenPose. The points (2,3),(3,4) form limbs of the right arm while the points (5,6),(6,7) form limbs of the left arm.

each finger has 4 limbs -

Normalizing limb lengths

Different subjects are of different heights, and at different distances from the camera. In order to ensure height differences don't play a role in classification, normalisation was done. The average distance between the subject's eyes were chosen as a measure of length, and this was used normalize the position vectors of all limbs on the body. Eyes were chosen because OpenPose predicts eye positions

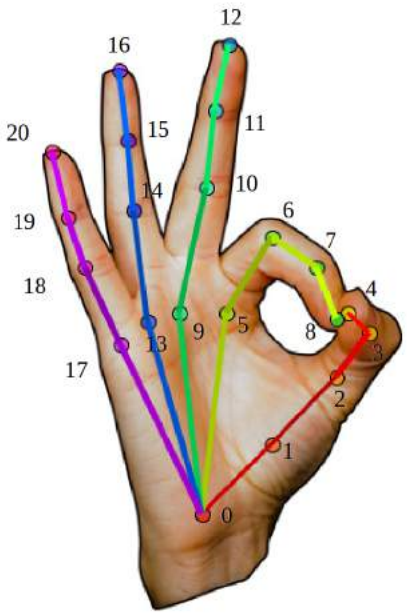


Figure 5.8: Skeletal points on each hand.Each finger has 4 limbs. For example, points (0,1),(1,2),(2,3),(3,4) make up 4 limbs for the thumb.

with maximum confidence, amongst all points in the body.

Characterising limb movement

Apart from limb, we are also interested in the movements of each limb. Hence, we also define limb velocity (as a measure of motion) and limb acceleration (to get further insights into motion) for each limb. Formally, velocity vector $\mathbf{V}_{l,t}$ is defined as (given a frame t and previous frame (t-1):

$$\mathbf{V}_{l,t} = \mathbf{D}_{l,t} - \mathbf{D}_{l,t-1}$$

Similarly, we define acceleration $\mathbf{A}_{l,t}$ as:

$$\mathbf{A}_{l,t} = \mathbf{V}_{l,t} - \mathbf{V}_{l,t-1}$$

For a video of N frames,we get N-1 velocity vectors per limb, and N-2 acceleration vectors per limb. Finally, we combine these 3 descriptors to form the features of a limb. Adjacent limbs in a finger/arm are also concatenated, to finally give us finger/arm vectors. We have 10 finger vectors (5 for each hand) and 2 arm vectors.

5.2.3 Visualising limb movements

Now that we have defined limbs, we need to see if these limbs exhibit certain properties, or whether they tend to form clusters. In order to get a better intuitive feel for what's happening, we do feature reduction using tSNE. We go from the original high dimensional space (which is 132 dimensional) to a 2D space, and visualize what we see. We also try to individually visualize each limb, by running tSNE separately for each limb. We see that the data points seem to form clusters.

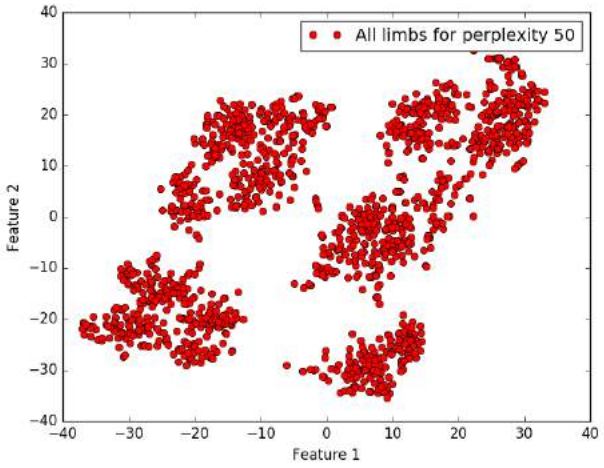


Figure 5.9: Plot of the feature space (all limbs) after tSNE

Each data point in the plot represents one frame in a video.

If we observe a plot of a single limb, we see clusters forming in that as well. This implies that the different actions being performed for each sign seems to be composed of a few "action states". We use these actions states to further reduce

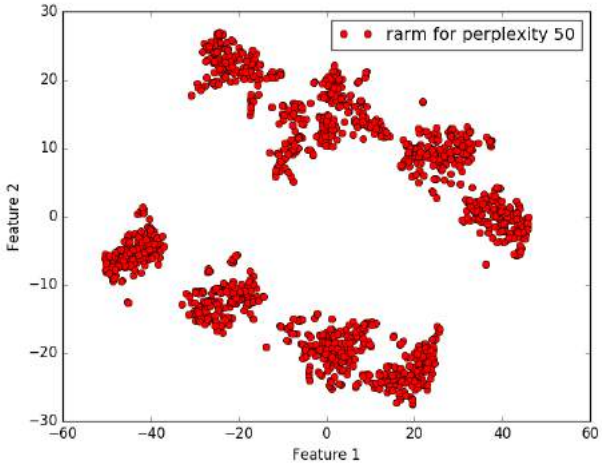


Figure 5.10: Plot of the feature space (right arm) after tSNE

our feature space.

5.2.4 Defining limb states

We have observed that each limb seems to form clusters. We now run k-Means on each limb, in order to label these various clusters.

In order to choose the right number of clusters,we rely both on visualisation, as well as the Silhouette score for each cluster. We observe that the Silhouette score often does not show large variations as we change the number of clusters. For each limb, we find the optimal number of clusters required, to represent various action states. The results are shown in table 5.2. We observe that most limbs exhibit 5 states. Some of these states have intuitive interpretations. For example, the 5th state in the right arm corresponds to frames which show the right arm at rest (no movement). However, other states do not seem to have intuitive explanations, and cannot be esily discerned by observing their corresponding video frames. We now try to observe how these states change with time, and whether states of certain limbs are correlated with states of other limbs.

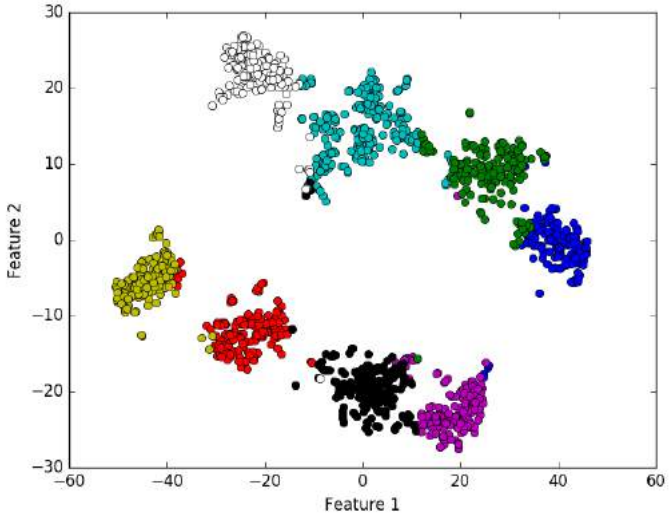


Figure 5.11: Running kMeans on the right arm. The clusters have been formed in the high dimensional space, with the 2D space just being used for visualisation.

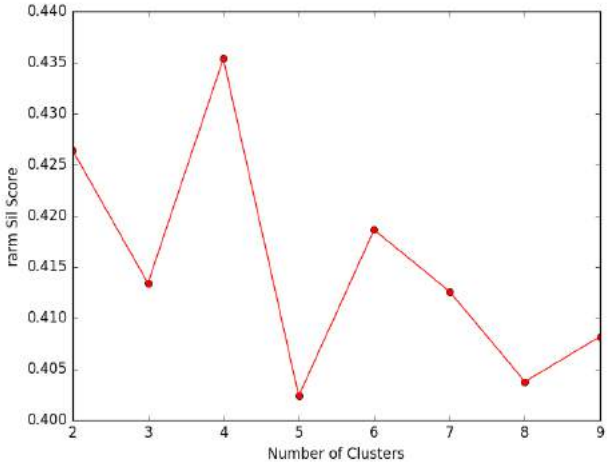


Figure 5.12: Variation of Silhouette score with number of clusters. We observe that that the value varies very little with number of clusters (0.4-0.435)

Table 5.2: Number of clusters for every limb

Limb	Number of clusters
rthumb	5
rindex	5
rmiddle	5
rring	5
rlittle	5
lthumb	5
lindex	5
lmiddle	5
lring	4
llittle	3
rarm	5
larm	5

Temporal relation between limb states

We first observe how limbs transition from one state to another. To do this,we build a state transition matrix for the right arm. After filtering out small probabilities (less than 0.1), we observe that each states branches out to a maximum of 2 other states, and they tend to form loops with each other. In the state transition graph for the right arm (Figure 5.13), we observe that State 2 is a central node. The states 0,4,2 form a loop, as well as states 1 and 2, and states 1,3 and 2. States 0 and 3 also form self loops, which indicates that consecutive frames of a video have a high possibility of being in the same state.

Another thing that can be observed is that these state transition diagrams are significantly different for different words. By comparing the transition diagrams for the words City and Street, we observe that the transitions between states and their probabilities vary significantly between the two. This can be used as a way to do classification as well. We also observe and compare their histograms, to see the number of times each state is visited for the 2 words. The histograms can be

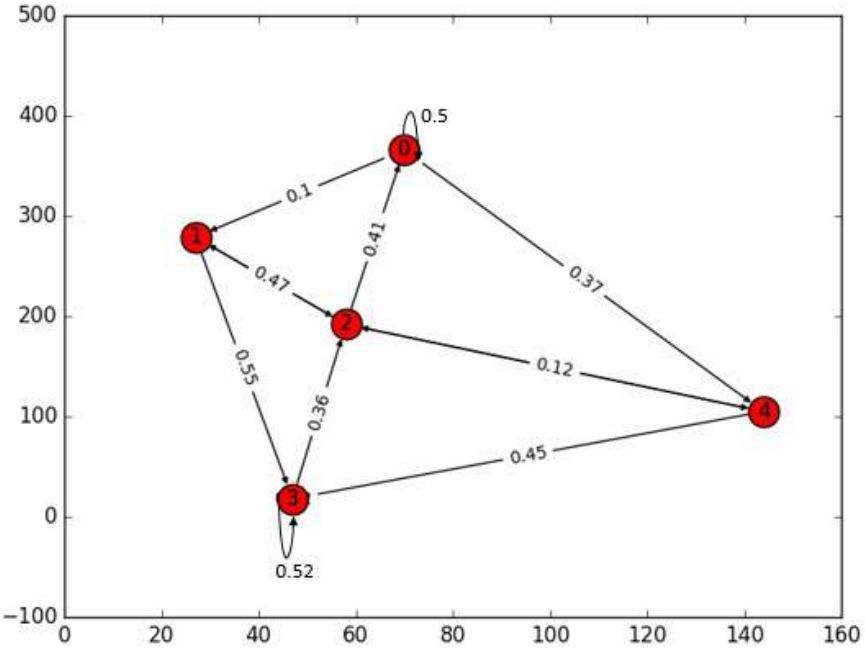


Figure 5.13: State Transition Diagram for the right arm, for the word city.

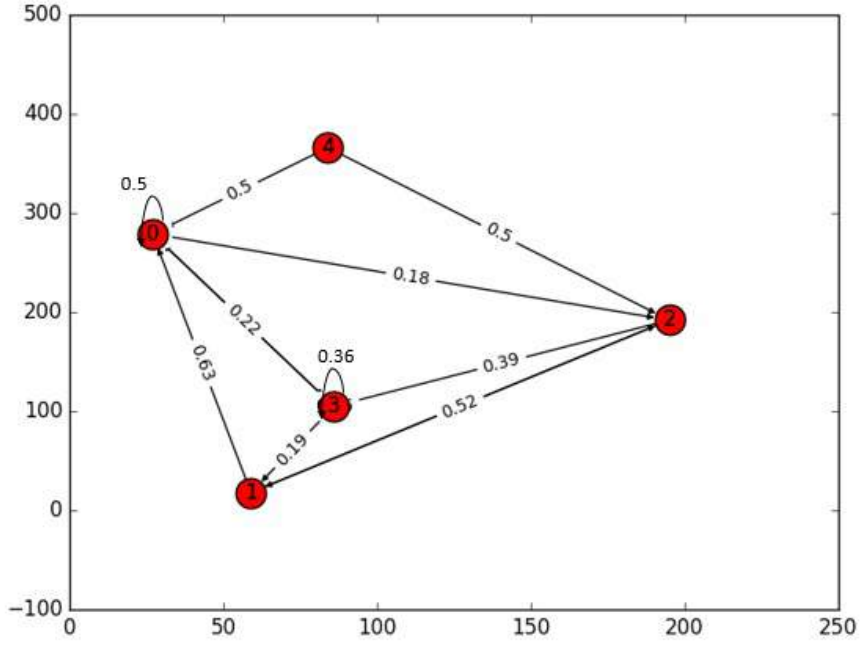


Figure 5.14: State Transition Diagram for the right arm, for the word Street. The transitions and their probabilities are significantly different from that for the word City.

used as a feature during classification as well. Howevver, not all words show such

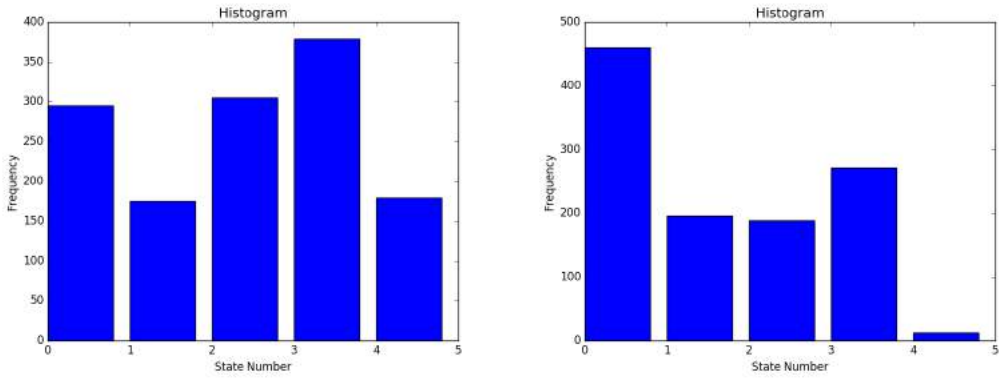


Figure 5.15: Histograms for the words City (Left) and Street (Right)

dinstinct transition graphs. The words City and House for example, have pretty similar transitions.

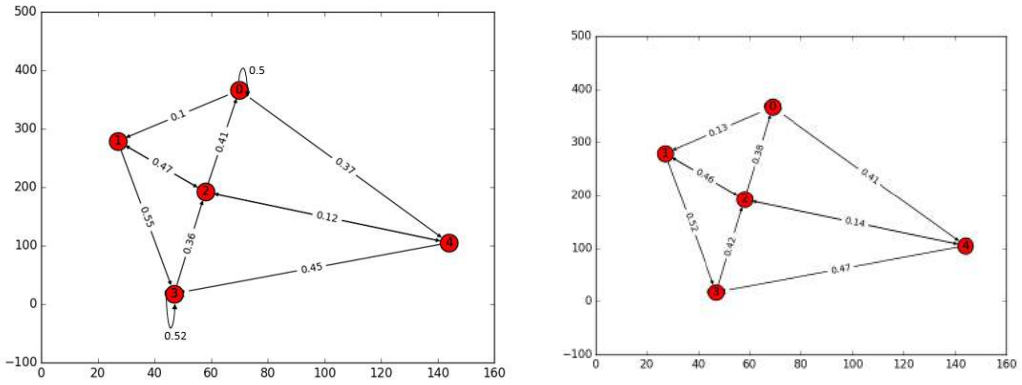


Figure 5.16: Transition Graphs for the words City (Left) and House (Right). Both graphs are pretty similar, with slightly different transition probabilities

Spatial relation between limb states

We also look at the correlation between various limbs in a single frame. As these states are discrete variables, we use Cramer's V Coefficient as a measure of correlation.

We observe that degree of correlation between various limbs varies significantly from one sign to another. However, certain general trends emerge:

- There seems to be a large degree of correlation (around 0.6-0.95, depending on the word) between the index, middle and ring fingers of each hand.
- The left and right arms show a large degree of correlation. This is possibly because each sign starts with the arms in resting position, and ends with the arms coming back to the same resting position.
- The thumbs do not show a high degree of correlation with any other fingers.
- There are large variations in these correlations from one sign to another.

Based on these observations, it may be possible to use limb correlations in testing data to try and classify signs as well.

CHAPTER 6

CONCLUSION AND FUTURE WORK

We see that this project is an attempt to translate ISL using an ensemble of techniques. Below, we quickly summarize the progress so far:

- India's first publicly available ISL dataset has been created. This solved a major problem in the space, as the lack of a dataset hindered any progress in ISL translation.
- We have observed that individual images of ISL poses can easily be dinstinguished by using OpenPose and a Decision Tree classifier.
- To aid video classification, we define features (limbs) which take into consideration the relative position of important points, as well as temporal information like their velocity and acceleration.
- These limbs exhibit certain patterns and form clusters. These clusters are then grouped into states, which reduce the feature space.
- These states have unique temporal and spatial properties for a given sign, and classification can be done by using state transitions, state histograms or their limb correlations as well.

Given this, there is still some way to go before we can fully classify all words. The following are the steps that should be taken up in the future:

- Keypoints should be extracted for all videos in the dataset. This is a long process which will take considerable time.
- Once all these keypoints have been obtained, clustering can be done on each limb to obtain their states.
- Once these states have been obtained, the corresponding state transition matrices for each word can be obtained and used for training a classifier.
- Alternatively, other approaches can be tried like selecting a subset of frames for each video, which can be used to represent the video instead. We have seen that single image classsification is easy,and hence if a small subset of frames can be accurately chosen, classification can be done in this way as well.

- We currently have 20 videos per sign.More data collection will help improve model accuracy and help prevent overfitting.

APPENDIX A

APPENDIX A: Banking Dataset Words

Below is a list of words present in the Banking Dataset, along with the number of videos recorded per word.

Table A.1: List of banking words and number of videos per word.

Banking Dataset	
Word	Number of Videos
Abbreviation	10
Accessories	10
Adjustment	10
Allowance	10
Assistance	12
Balance	10
Benefit	13
Bill	5
Borrower	10
Business	10
Calculation	10
Capital	10
Cash	12
Company	10
Complaint	10

Compound Interest	10
Compulsory	10
Conference	10
Counter	10
Courier	10
Declaration	10
Difference	10
Documents	10
Economic	10
Emergency	10
Expiry	10
Fake	10
Fixed Rate	10
Free	13
Growth	10
Holder	10
Identification	10
Illegal	10
Incentive	10
Income	10
Information	11
Interest	10
Job	15
Judicial	10
License	10

Monthly return	10
Official	10
Parent Branch	9
Partner	10
Password	10
Penalty	12
Quantity	10
Quotation	10
Recover	10
Register	10
Reservation	10
Risk	10
Tender	10
Thank You	12
Unconditional	10
Urgent	12
Wealth	10

APPENDIX B

APPENDIX B: General Dataset Words

Below is a list of words present in the General Dataset, along with the number of videos recorded per word.

Table B.1: List of words and number of videos per word.

Word	Number of Videos
Animals	
Dog	20
Cat	20
Fish	20
Bird	25
Cow	21
Mouse	20
Horse	20
Animal	20
Transportation	
Train	21
Plane	21
Car	20
Truck	21
Bicycle	20
Bus	20

Boat	21
Train Ticket	21
Transportation	21
Location	
City	20
House	21
Street/Road	20
Train Station	22
Restaurant	20
Court	23
School	20
Office	20
University	21
Park	22
Store/Shop	22
Library	20
Hospital	20
Temple	21
Market	21
India	21
Ground	21
Bank	22
Location	22
Clothing	
hat	20

Dress	20
Suit	19
Skirt	19
Shirt	20
T-Shirt	20
Pant	20
Shoes	20
Pocket	20
Clothing	20
Colour	
Red	20
Green	20
Blue	20
Yellow	20
Brown	21
Pink	20
Orange	20
Black	20
White	20
Grey	21
Colour	20
People	
Son	20
Daughter	20
Mother	20

Father	20
Parent	20
Baby	20
Man	20
Woman	20
Brother	21
Sister	20
Family	17
Grandfather	17
Grandmother	17
Husband	20
Wife	20
King	20
Queen	20
President	20
Neighbour	20
Boy	21
Girl	20
Child	20
Adult	20
Friend	20
Player	20
Crowd	20
Electronics	
Clock	13

Lamp	14
Fan	15
Cell Phone	14
Computer	14
Laptop	14
Screen	14
Camera	14
Television	14
Radio	14
Home	
Table	14
Chair	14
Bed	14
Dream	14
Window	14
Door	14
Bedroom	14
Kitchen	14
Bathroom	14
Pencil	14
Pen	14
Photograph	14
Soap	14
Book	14
Page	14

Key	14
Paint	15
Letter	14
Paper	14
Lock	14
Telephone	14
Bag	14
Box	14
Gift	14
Card	14
Ring	14
Tool	14
Jobs	
Teacher	14
Student	14
Lawyer	14
Doctor	14
Patient	14
Waiter	14
Secretary	14
Priest	15
Police	14
Soldier	14
Artist	14
Author	14

Manager	14
Reporter	14
Actor	14
Job	14
Seasons	
Summer	14
Spring	14
Winter	14
Fall	15
Season	14
Monsoon	14
Society	
Religion	14
Death	14
Medicine	14
Money	14
Bill	14
Marriage	14
Team	14
Race(ethnicity)	14
Technology	14
Energy	14
War	14
Peace	14
Attack	15

Election	14
Newspaper	14
Gun	14
Sport	14
Exercise	14
Ball	15
Price	14
Sign	14
Science	14
God	14
Time	
Sunday	14
Monday	14
Tuesday	14
Wednesday	14
Thursday	14
Friday	14
Saturday	14
Today	14
Tomorrow	14
Yesterday	14
Week	14
Month	14
Year	15
Hour	14

Minute	14
Second	15
Morning	14
Afternoon	14
Evening	14
Night	15
Time	15
Adjectives	
Loud	21
Quiet	21
Happy	21
Long	21
Short	22
Tall	21
Wide	21
Narrow	21
Big (large)	21
Small (little)	20
Slow	21
Fast	21
Hot	21
Cold	20
Warm	21
Cool	21
New	21

Old	21
Young	21
Good	21
Bad	21
Wet	21
Dry	21
Sick	21
Healthy	21
Sad	8
Beautiful	9
Ugly	8
Deaf	8
Blind	8
Nice	4
Mean	8
Rich	8
Poor	8
Thick	8
Thin	4
Expensive	8
Cheap	8
Flat	8
Curved	8
Male	8
Female	8

Tight	8
Loose	8
High	8
Low	8
Soft	8
Hard	8
Deep	8
Shallow	8
Clean	8
Dirty	8
Strong	8
Weak	8
Dead	8
Alive	8
Heavy	8
Light	8
Famous	8
Pronouns	
I	21
You	21
He	21
She	21
It	21
We	21
You (plural)	21

They	21
Greetings	
Hello	21
How are you	21
Alright	21
Good morning	21
Good afternoon	22
Good evening	21
Good night	21
Thank you	21
Pleased	21

REFERENCES

(). Segment, track, extract, recognize and convert sign language videos to voice/text. *International Journal of Advanced Computer Science and Applications*.

ANSARI, Z. A. and **G. HARIT** (2016). Nearest neighbour classification of indian sign language gestures using kinect camera. *Sadhana*, **41**(2), 161–182. ISSN 0973-7677. URL https://doi.org/10.1007/s12046-015-0405-3.

Cai, X., **W. Zhou**, **L. Wu**, **J. Luo**, and **H. Li** (2016). Effective active skeleton representation for low latency human action recognition. *IEEE Transactions on Multimedia*, **18**(2), 141–154.

Cao, Z., **G. Hidalgo**, **T. Simon**, **S. Wei**, and **Y. Sheikh** (2018). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, **abs/1812.08008**. URL http://arxiv.org/abs/1812.08008.

Rao, G. A. and **P. Kishore** (2018). Selfie video based continuous indian sign language recognition system. *Ain Shams Engineering Journal*, **9**(4), 1929 – 1939. ISSN 2090-4479. URL http://www.sciencedirect.com/science/article/pii/S2090447917300217.

Singha, J. and **K. Das** (2013). Recognition of indian sign language in live video. *CoRR*, **abs/1306.1301**. URL http://arxiv.org/abs/1306.1301.

van der Maaten, L. and **G. Hinton** (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.