ISYE 6740 – Spring 2021

Deepthi M Rao, Harikiran Cherala

SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING

Problem Statement

Sign Language has been a very important source of communication among deaf people throughout history. It is a type of communication that is conveyed through a mixture of gestures performed by our hand, head and body. Now with the advent of various technologies like sensors, machine learning, computer vision, communication for deaf people has become easier where there can communicate with people who do not know the sign language and the technology helps in the translation of the language. There are many different types of sign languages that have been evolved throughout the world. We will be focusing our project on Indian Sign Language (ISL).

ISL uses a combination of both hands to convey the letters of the English alphabet from A to Z. Our project aims to build a machine learning algorithm that recognizes and classifies the Indian Sign Language visually from static images showing various hand gestures representing the English alphabet and numbers with the help of various ML algorithms.

Data Source

The data consists of 1200 images per letter of the alphabet of size $128x128$. It consists of cropped images of hands showing the sign for each letter. This data was captured ensuring constant illumination in the background.

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

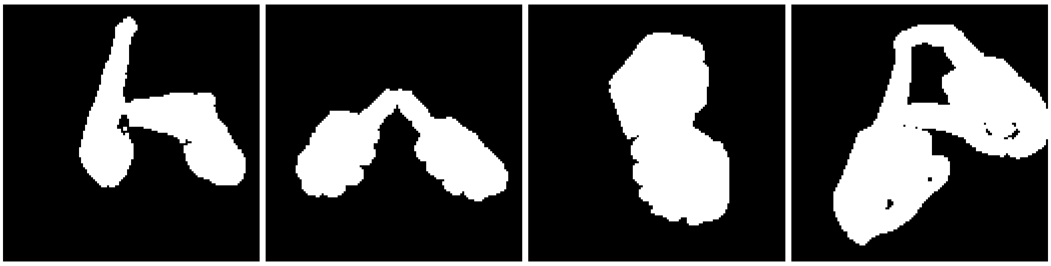
Z

Methodology Proposed

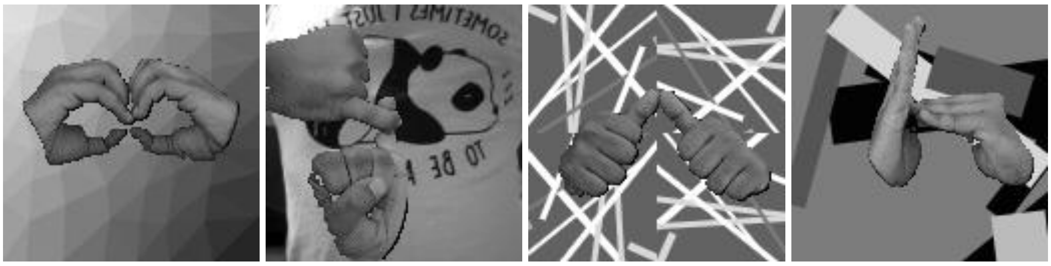
Data Acquisition and Pre-processing

The above given data has a constant background across all classes. Training a multi-class classification problem on such kind of data would lead to biased results. Hence, we perform synthetic data generation by performing segmentation and adding different backgrounds. Segmentation was done to extract only the palms from the image by using a method of skin segmentation where the RGB image is converted to HSV image, and a range is given for the colour of skin. Based on this range, a binary mask is constructed where all the white pixels indicate the palm or skin and the black pixels are the background. Once the mask is obtained using a series of bitwise operations, we add multiple backgrounds randomly taken from the Internet to the data and construct the data set.

The images below show some binary images after skin segmentation is performed using HSV method.

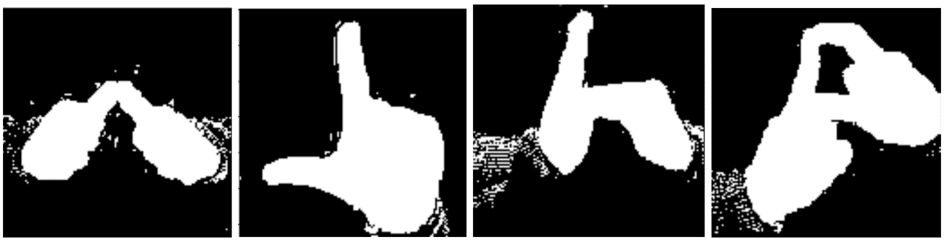


The images below show some letters after performing segmentation and adding random backgrounds.



Skin Segmentation can also be done with the help of Gaussian mixture models (GMM). GMM tries to approximate the distribution of data using a mixture of gaussian distributions. The parameters for maximum likelihood learning for GMM can be represented as $\theta = (\pi_\mathrm{k}, \mu_\mathrm{k}, \Sigma_\mathrm{k})$ where π_k represents the weight the kth gaussian component contributes and $\mu_\mathrm{k}, \Sigma_\mathrm{k}$ are the mean and covariance of the kth gaussian component. GMM can be used for skin segmentation as it tries to classify each pixel into k components or classes and hence cluster the pixels into skin and not skin. This will be particularly helpful in our dataset as the skin and the background can be easily differentiable. The mean vector μ_k was initialized by using K-means clustering algorithm. The cluster centres obtained from K-means was used as initialization for the mean vector.

The images below show binary images after skin segmentation is performed using GMM method.

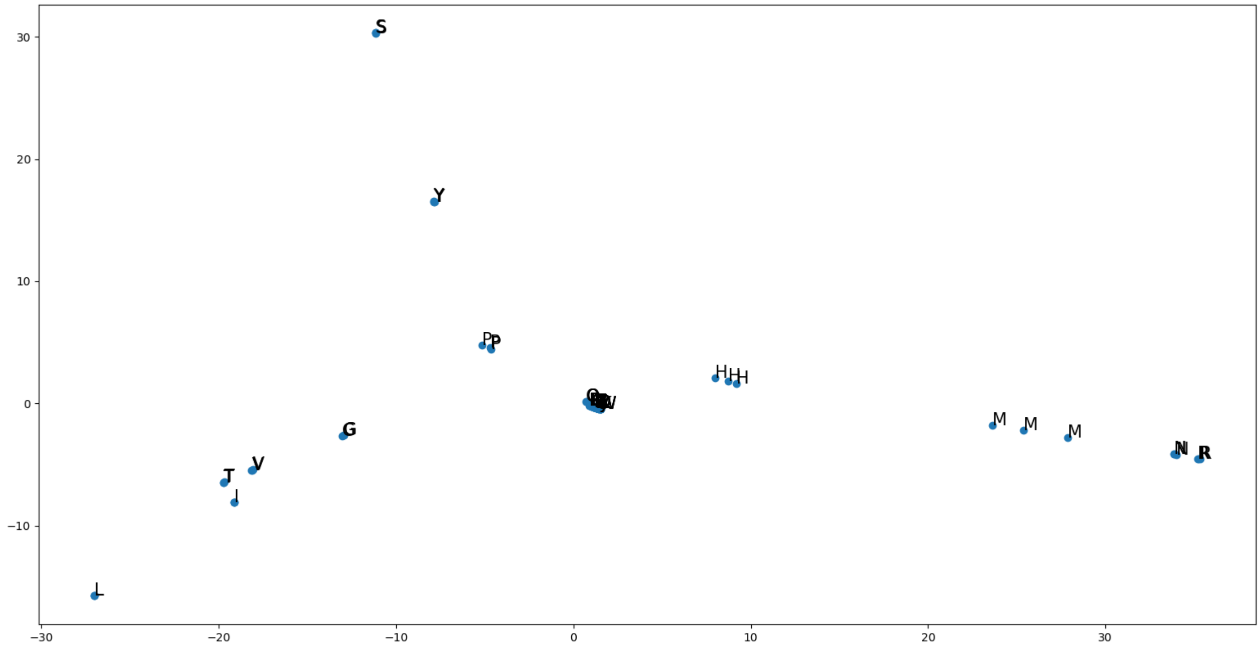


Each image was resized down to 64×64 and converted to grayscale after pre-processing and is vectorized and stored in an n-dimension array, where each row represents the vectorized image and the columns are the pixels.

Feature Extraction

Feature extraction is a method of extracting important parts of data and representing it mathematically in the form of a vector or matrix. There are many types of feature extraction available. We used the following methods to extract features: -

- **Principal Component Analysis (PCA) –** PCA is a linear dimensionality reduction technique that identifies important features or relationships in our data by finding components in the direction where maximum variance can be achieved and then transforms our data based on the top n components. The covariance matrix of our data is calculated and eigen decomposition is performed on this matrix and we get eigenvalues and eigenvectors. These eigenvectors are used to transform our data. For our purpose we chose top 20 components.
- **ISOMAP –** ISOMAP is a non-linear dimensionality reduction technique. We try to produce a low dimensional representation which preserves the “walking distance” over the manifold (data cloud). An adjacency matrix is constructed based on nearest neighbours’ method and then the shortest path graph is calculated. A centering matrix is calculated and eigen decomposition is performed to get the components. We down sampled the dataset and chose 2 components and performed ISOMAP and plotted the transformed data. The points formed in the scatter plot were clustered based on the label. Each label is represented by some orientation of the hand, ISOMAP is able to learn this orientation. We performed K-Means clustering on the ISOMAP transformed data to cluster the data points.



Multi-class Classification

This is a multi-class classification problem with 26 classes representing each letter of the alphabet. The data was split into 80% training and the rest 20% was used for testing. Multiple ML algorithms for classification was used on the training data with and without performing PCA. The ML algorithms used are: -

- **Logistic Regression** – It is a classification algorithm that uses the logistic function (an S-shaped curve) to model the dependant variable. It uses the stochastic gradient decent algorithm to solve the optimization problem. We added a maximum iteration of 10 and the results were evaluated.
- **K-Nearest Neighbours** – It is a supervised ML algorithm, that classifies the points based on its K-nearest neighbours and performs a majority vote. The distance metric used is Euclidean and we used 3 as the nearest neighbour.
- **Support Vector Machine (SVM)** – This is a supervised ML algorithm where a decision boundary needs to be identified based on the data. The decision boundary can either be linear or non-linear. We tried both linear and non-linear kernel (Radial basis function kernel).
- **Multi-Layer Perceptron (MLP) –** 1 hidden layer with 100 neurons was used with initial learning rate of 0.1.
- **Decision Tree Classifier –** A decision tree is constructed based on the features and classification is performed.
- **Random Forest Classifier –** This is a classifier where multiple trees are used and then a majority vote is taken to perform classification. Total number of trees in the forest were chosen to be 100.
- **Linear Discriminant Analysis –** We try to fit multivariate normal distribution to the data and get a decision boundary which is linear. This happens when the co-variance matrices of all the normal distributions are equal.
- **Quadratic Discriminant Analysis –** Here we try to fit multivariate normal distributions with different co-variance matrices making the decision boundary non-linear.
- **Gaussian Naïve Bayes –** Here a gaussian distribution is fit to the data assuming that all the features are independent of each other.

Evaluation Metrics

Confusion Matrix: - It is an N x N matrix, where N denotes the total number of classes in the classification problem. It Consists of true positives, false positives, true negatives and false negatives.

Mis-classification error: - It is defined as the ratio of the total number of mis- classified samples to the total number of classifications.

Precision: - It is defined as the ratio of the total number of correctly classified instances to the total number of retrieved instances. TP – True Positives FP – False Positives TN -True Negatives, FN – False Negatives.

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

Recall: - It is defined as the ratio of the total number of correctly classified instances to the total number of relevant instances.

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

F1-score: - It is defined as the harmonic mean of precision and recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Accuracy: - It is defined as the ratio of the total number of correct predictions to the total number of input samples.

Results

The following table shows the accuracies achieved for all the models on the testing data after performing PCA.

Model	Accuracy (%)
Logistic Regression	52
K-Nearest Neighbours (k=3)	98
SVM with linear kernel	82
SVM with radial basis function kernel	99
Multi-layered perceptron	99
Random Forest Classifier	99
Decision Tree Classifier	95
Linear Discriminant Analysis	48
Quadratic Discriminant Analysis	93
Gaussian Naïve Bayes	48

The following table shows the accuracies achieved for the models on the testing data without PCA.

Model	Accuracy (%)
Logistic Regression	96
K-Nearest Neighbours (k=3)	99
Random Forest Classifier	99
Decision Tree Classifier	96
Linear Discriminant Analysis	98
Quadratic Discriminant Analysis	34
Gaussian Naïve Bayes	97

The precision, recall and F1-scores for one of the models – Logistic Regression for data trained without applying PCA is given below: -

Class	Precision (%)	Recall (%)	F1-score (%)
A	93	94	94
B	90	93	91
C	88	95	92
D	99	99	99
E	99	95	97
F	99	99	99
G	88	99	98
H	99	100	100
I	81	90	85
J	99	99	99
K	99	98	99
L	97	90	94
M	99	100	100
N	99	99	99

O	85	89	87
P	97	97	97
Q	100	100	100
R	100	97	98
S	97	96	97
T	99	94	96
U	94	95	94
V	88	88	88
W	96	96	96
X	98	91	95
Y	98	99	99
Z	99	99	99

Conclusion and Future Work

We can see that the for the classification of ISL data for static images, non-linear SVM, MLP and Random Forest classifier performed well for data after performing PCA with an accuracy of 99%, whereas for data without performing PCA, Random Forest Classifier and KNN performed well with an accuracy of 99% followed by LDA, decision tree and Logistic Regression.

The future work would involve collecting more data with different backgrounds and variations and manually annotating it and train using multiple models. We can also build a convolutional neural network to localize the hands in the image and classify the gesture for better accuracies and it can be used to test on video data. LSTMs and other deep neural networks can be experimented with to perform classification for words which involve a series of actions.

WORK SPLIT UP

Data Acquisition and Pre-processing -
Skin Segmentation with HSV – Harikiran
Skin Segmentation with GMM – Deepthi
Pre-processing – Harikiran and Deepthi

Feature Extraction -
PCA – Harikiran
ISOMAP – Deepthi

Classification and Results – Harikiran and Deepthi