CS5710: Machine Learning
Project Proposal + Increment

Project Title: Hand Gesture Alphabet Sign Language Recognition

Team Members

1. Venkata Krishna Lakshmi Spandana Vegi – 700744287 (vxv42870@ucmo.edu)
2. Sindhu Rajanala – 700741228 (sxr12280@ucmo.edu)
3. Sai Swetha Nambari –700742846 (sxn28460@ucmo.edu)
4. Venkata Lakshmi Sasank Tipparaju – 700738838 (vxt88380@ucmo.edu)

Advisor: Prof. Muhammad Zubair Khan.

Motivation

In our modern world, it's necessary to socialize with all kinds of people either to gain knowledge on various aspects or to improve their communication skills. As per World Health Organization, there are around 5% of people who are deaf i.e., around 700 million. Of which 63% are deaf by birth and others were losing their hearing ability when they met with accidents [1]. It's the similar case with speech disability people. But as the world advances, they can communicate within each other with sign languages such as hand gestures. This Sign language is like a bridge that connects the hearing disabled and silent/speech disability people.

In Society, communication is one of the major necessities to live on, but in our daily life we see that normal people are facing issues while communicating with deaf and dumb people. The sign language used to communicate, that includes actions like bodily gestures, face expressions, hand gestures, etc. are difficult to understand for the normal people. However, people who understand hearing and speech disabled are very limited to communicating with hand signs. People get depressed when they are not able to communicate with their loved ones who have speech/hearing disability.

Objective

In order to help the deaf-mutes and normal people to be able to communicate easily and understand each other, There is a need for an application which can help in analyzing and decode the hand gestures to produce data that can be understandable by all. Hence, we are going to implement machine learning prediction algorithms – Logistic Regression, Decision Tree, Support Vector Machine, Random Forest on the Sign Language datasets to check the accuracy of each algorithm and provide analysis on the performance metrics

Significance:

We are going to build a sign language recognition system with the help of machine learning algorithms so that this system can help the normal people by capturing all their hand gestures and process them to be able to predict the context of what they are trying to communicate and conveys the same information to the end user.

There are already many researchers working on the same sign language prediction systems with various machine learning algorithms from the past decade.

In our work we are going to experiment the implementation of this Sign Language System with multiple machine learning algorithms and compare them in terms of performance metrics - with accuracy & efficiency in the outcomes of the prediction algorithm i.e., accuracy that it predicts correct meaning to the hand questers

Related Works

“A New Benchmark on American Sign Language Recognition using Convolutional Neural Network"(M. M. Rahman et. al.,2019). The above-mentioned work gives us an idea on the detection of American Sign Language by using convolutional neural networks. About four datasets were considered with good reports. The performance of the proposed model is studied on each dataset when trained and tested. The model has an accuracy of 100% while recognizing both digits and alphabets and it has an accuracy of 99.90% with the digit and the sign language [2].

"ML Based Sign Language Recognition System" (K. Amrutha and P. Prabu,2021) This research talks about automated identification of SLR based on vision-based isolated hand gesture detection and recognition utilizing convex Hull feature extraction and KNN as classifier which yielded an 65% accuracy [3].

” Indian Sign Language recognition system using SURF with SVM and CNN” (Shagun Katoch et. at., 2022). In this work the Support Vector Machine and Convolutional Neural Networks are used for the classification. The training data used is 80% and 20% of the data is used for the purpose of testing. For classification SVM with linear kernel is used. SVM has given an accuracy of 99.14% on test data and overall accuracy of 99%. Whereas CNN has given an accuracy of 99% on testing data and overall accuracy of 94% on training data [4].

Methodology

Here we adopt the sequential method of machine learning which means which can import or export the data sequence like text streams, audio clips, video clips, time-series data, etc., In this method we train the data by the help of the CNN algorithm and obtain accuracy and then later we implement the methods of the machine learning.

First, we will collect the data with all hand gestures images files and categorize them with labels either alphabets or some words so that on passing the image as input the respective context will result as output. On the collected dataset, we divided our approach to tackle the classification problem into three stages. The first stage is to segment the skin part from the image, as the remaining part can be regarded as noise w.r.t the character classification problem. The second stage is to extract relevant features from the skin segmented images which can prove significant for the next stage i.e., learning and classification. The third stage as mentioned above is to use the extracted features as input into various supervised learning models for training and then finally use the trained models for classification.

In the second step we will augment the data and preprocess it before sending it to any of the classifiers. In the third step we will split the data into test and training in-order for the algorithm results to be efficient. Then depending on the dataset size, we will decide whether we must do cross folding or not. Because if our dataset is too large and then if we do cross folding it may lead to overfitting. We will take training data and perform feature extractions, image processing etc. and then we will process it into the respective classifier. We will use the following Machine Learning algorithms to implement the Sign Language System

Proposed Methods

The respective process is clearly described in the given flow chart

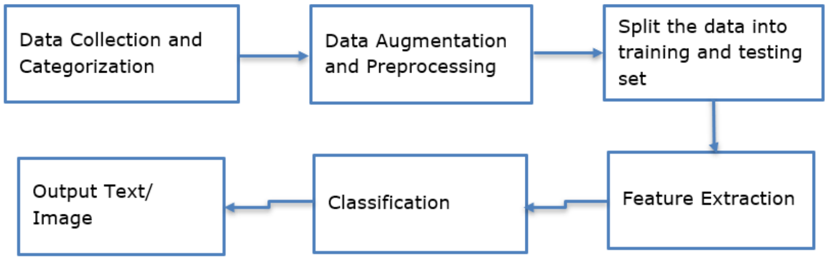

Data Collection and Categorization
Data Augmentation and Preprocessing
Split the data into training and testing set
Feature Extraction
Classification
Output Text/ Image

Fig 1.a

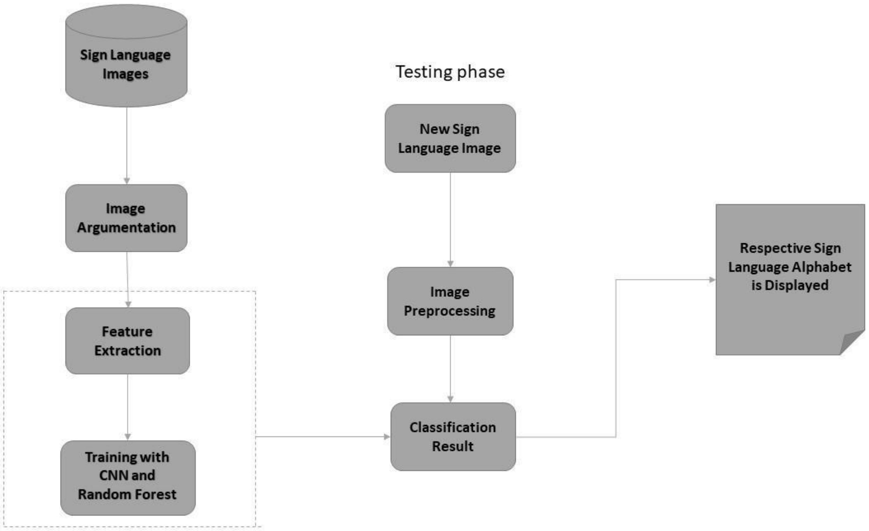


Fig 1.b

1.1.Dataset

We have downloaded datasets from Kaggle which are open source for sign language recognition systems. Dataset consists of the American Sign language which utilizes one hand gestures to communicate with each other. There are images resembling each alphabet which are the count of the 26 and one additional folder of images for the unveiled hand gestures gives a total of 27 files. The Total data set size consists of 40,000+

images in ".jpg' format. we separate 500 images from each primary folder and make another secondary folder, where the primary folder is used to train the data and secondary folder is used to test the models

1.2.Image Segmentation:

The steps involved in the image segmentation are image resizing, converting all the images of different sixes to the same size i.e., (128,128). Data scaling where we scale the data to the model accordingly. Data transfer converting each pixel data to the data frame

Image layering:

- Convolutional layers in a convolutional neural network summarize the presence of features in an input image. As of now, we use 4 conventional layers
- Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. We use 4 pooling layers
- The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. We use 4 dropout layers
- Dense Layer is a simple layer of neurons in which each neuron receives input from all the neurons of the previous layer. We use 4 dense layers.
- A flatten layer collapses the spatial dimensions of the input into the channel dimension. For example, if the input to the layer is an H-by-W-by-C-by-N-by-S array (sequences of images), then the flattened output is an (H*W*C)-by-N-by-S array. We use 1 flatten layer. In total we use 15 layers of image processing steps in our code.
- and we used an activation function (Relu)

1.3.Implementation Algorithms

1.3.1.Logistic Regression:

Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. The accuracy is gained in the logical regression is > 82 %, < 90 %.

1.3.2.Decision Tree:

Decision Tree is the most powerful and popular tool for classification and prediction. A decision tree is a flowchart-like representation of data that graphically resembles a tree that has been drawn upside down. The accuracy is gained in the logical regression is >85 %, <96 %.

1.3.3.Support Vector Machine:

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. Accuracy not gained

1.3.4.Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Accuracy not gained

2. Learning/Contributions:

Data Augmentations/Preprocessing, Splitting of Data using K Cross fold validation if required, Feature Extraction, Various Machine Learning Classifier algorithms like Decision Tree, Logistic Regression, SVM, Random Forest and Checking accuracy and performance metrics of each algorithm based on predictions of Sign Language. All the work will be divided equally within group members based on their interested part.

3. Evaluation/Results:

We will load all the training data and testing data into the classifier and check the accuracy. In the final stage we will deploy the system so that we can test it on real time scenarios. On deploying the application an html link is generated, on clicking it, the user will be asked to provide the input of image browsing from the local system. On providing the image, the sign language recognition system will process the input and decode the sign into readable text - in our case it will be alphabets since we categorize each image with an alphabet. And will compare the results based on most effective performance measurement Accuracy.

4. Dataset

We have downloaded datasets from Kaggle which are open source for sign language recognition systems. Dataset consists of the American Sign language which utilizes one hand gestures to communicate with each other. There are images resembling each alphabet which are the count of the 26 and one additional folder of images for the unveiled hand gestures gives a total of 27 files. The Total data set size consists of 40,000+ images in ".jpg' format. we separate 500 images from each primary folder and make another secondary folder, where the primary folder is used to train the data and secondary folder is used to test the models

5.Image Segmentation:

The steps involved in the image segmentation are image resizing, converting all the images of different sixes to the same size i.e., (128,128). Data scaling where we scale the data to the model accordingly. Data transfer converting each pixel data to the data frame

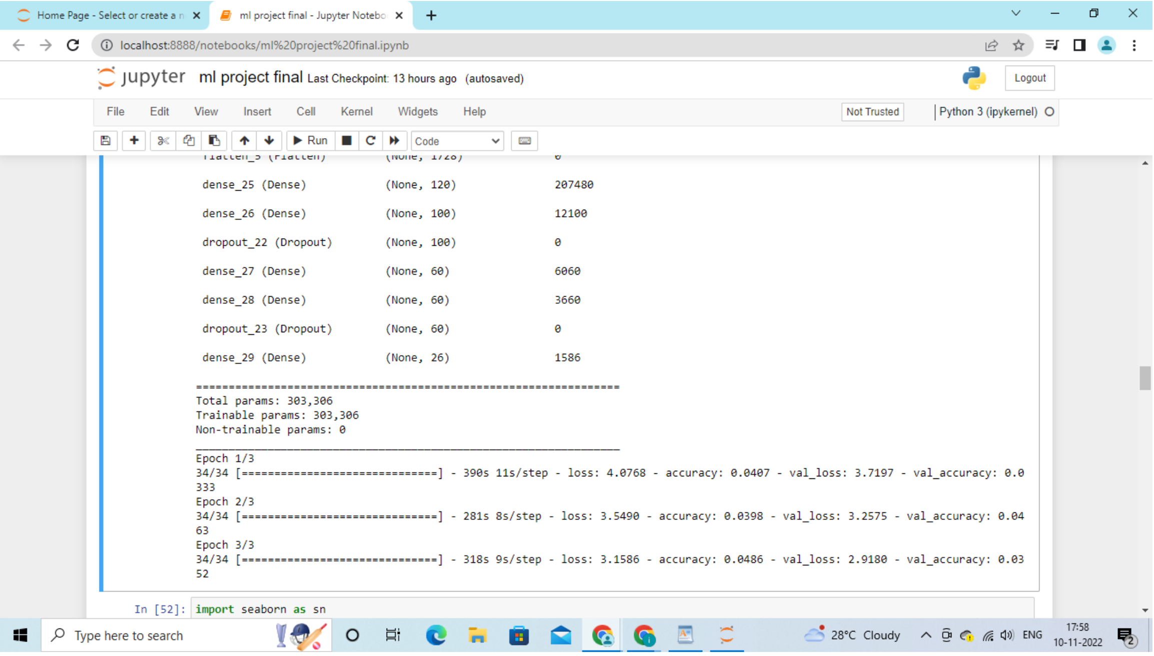
Image layering:

- Convolutional layers in a convolutional neural network summarize the presence of features in an input image. As of now, we use 4 conventional layers
- Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. We use 4 pooling layers
- The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. We use 4 dropout layers

- Dense Layer is a simple layer of neurons in which each neuron receives input from all the neurons of the previous layer. We use 4 dense layers.
- A flatten layer collapses the spatial dimensions of the input into the channel dimension. For example, if the input to the layer is an H-by-W-by-C-by-N-by-S array (sequences of images), then the flattened output is an (H*W*C)-by-N-by-S array. We use 1 flatten layer. In total we use 15 layers of image processing steps in our code.
- and we used an activation function (relu)

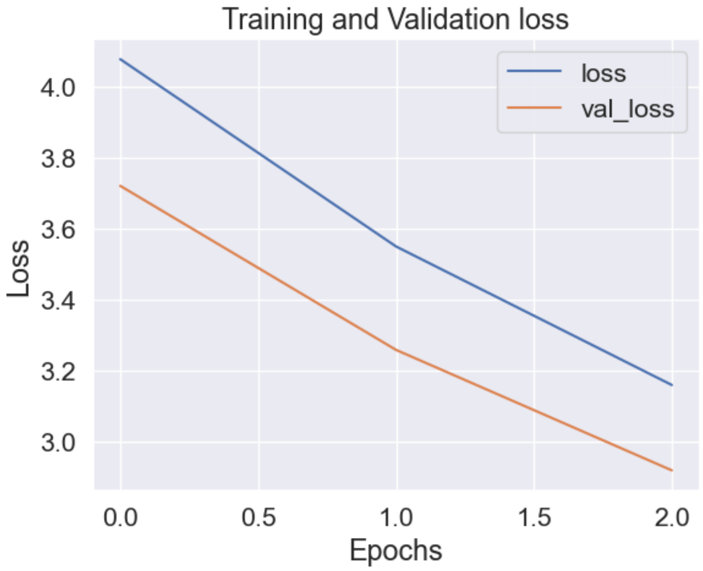
6.Preliminary Results

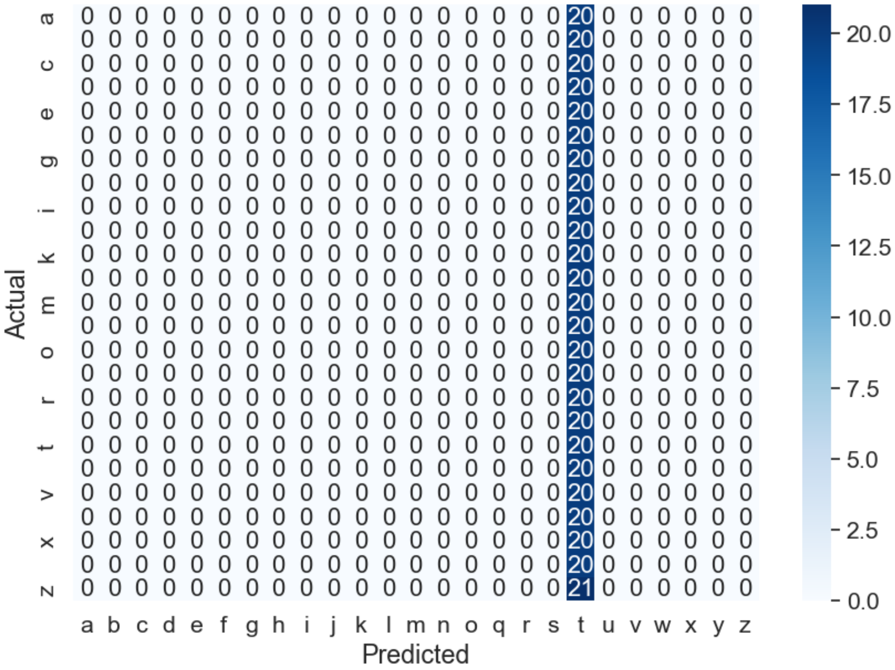
Sample data result


Home Page - Select or create a n
ml project final - Jupyter Notebo
localhost:8888/notebooks/ml%20project%20final.ipynb
jupyter ml project final Last Checkpoint: 13 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
28°C Cloudy
ENG
17:58
10-11-2022

```
 flatten_5 (Flatten)        (None, 1728)              0

 dense_25 (Dense)           (None, 120)               207480

 dense_26 (Dense)           (None, 100)               12100

 dropout_22 (Dropout)       (None, 100)               0

 dense_27 (Dense)           (None, 60)                6060

 dense_28 (Dense)           (None, 60)                3660

 dropout_23 (Dropout)       (None, 60)                0

 dense_29 (Dense)           (None, 26)                1586

=================================================================
Total params: 303,306
Trainable params: 303,306
Non-trainable params: 0
_________________________________________________________________
Epoch 1/3
34/34 [==============================] - 390s 11s/step - loss: 4.0768 - accuracy: 0.0407 - val_loss: 3.7197 - val_accuracy: 0.0
333
Epoch 2/3
34/34 [==============================] - 281s 8s/step - loss: 3.5490 - accuracy: 0.0398 - val_loss: 3.2575 - val_accuracy: 0.04
63
Epoch 3/3
34/34 [==============================] - 318s 9s/step - loss: 3.1586 - accuracy: 0.0486 - val_loss: 2.9180 - val_accuracy: 0.03
52
```

```
In [52]: import seaborn as sn
```


Training and Validation loss
loss
val_loss
Loss
4.0
3.8
3.6
3.4
3.2
3.0
0.0
0.5
1.0
1.5
2.0
Epochs


Actual
a
c
e
g
i
k
m
o
r
t
v
x
z
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 0 0 0 0 0 0
a b c d e f g h i j k l m n o q r s t u v w x y z
Predicted
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

7.Project management

7.1.Implementation Status Report

Work completed:

- Collected a dataset from kaggle and preprocessed the data by resizing, scaling and performing CNN layers to convert images pixel data to data frames.
- As the dataset contains around 40,000 sample of jpg files, we initially took 500 images files from each alphabet category and trained the data in machine learning algorithms such as Decision Tree and Logistic Regression which achieved a good accuracy rate around 80% - 90%
- Responsibility - as the dataset is huge and taking time to process, we divided into two groups and performed Decision tree and Logistic Regression algorithms training and validated accuracy. Provided screenshots in the preliminary results section
- We got the dataset in the format of images, so initially we struggled to process and fit them into machine learning algorithms, later on we utilized the CNN layering technique to process

7.1.1 Logistic Regression:

Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. The accuracy is gained in the logical regression is > 82 %, < 90 %.

7.1.2 Decision Tree:

Decision Tree is the most powerful and popular tool for classification and prediction. A decision tree is a flowchart-like representation of data that graphically resembles a tree that has been drawn upside down. The accuracy is gained in the logical regression is >85 %, <96 %.

7.1.3 Support Vector Machine:

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. Accuracy not gained.

7.1.4 Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Accuracy not gained.

7.2 Responsibilities & Contribution

Venkata Krishna Lakshmi Spandana Vegi	:	Topic selection, logistic regression.
Sindhu Rajanala	:	Decision tree, deep work on project proposal.
Sai Swetha Nambari	:	Dataset purification, dataset preprocessing.
Venkata Lakshmi Sasank Tipparaju	:	Raw data collection, deep work on project proposal, Documentation and reports for submission.

Work to be completed

- we need to perform training on other two algorithms and then start testing in all 4 algorithms and comparison of results with performance metrics
- Issues/Concerns Faced: In SVM and Random Forest we haven't achieved the good accuracy while training, need to check over the data preprocessing, scaling, cnn layering - image segmentation

8 Reference

1. World Health Organization WHO, [Online] Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

2. M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta and M. Aktaruzzaman, "A New Benchmark on American Sign Language Recognition using Convolutional Neural Network," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9067974. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9067974/citations#citations

3. K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), 2021, pp. 1-6, doi: 10.1109/ICITIIT51526.2021.9399594. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9399594

4. Shagun Katoch, Varsha Singh, Uma Shanker Tiwary, ”Indian Sign Language recognition system using SURF with SVM and CNN,” Array, Volume 14,2022,100141,SSN 2590-0056, https://doi.org/10.1016/j.array.2022.100141 (https://www.sciencedirect.com/science/article/pii/S2590005622000121)