Sign language Recognition System using Machine Learning

----------***----------
-

Abstract - *Sign language is a good visual communication aid for those with auditory disabilities. This language is also prevalent for those with speech impairment. However, the general populous have little knowledge on sign language, and often find difficulty communicating with someone who is primarily versed in sign language. ISL is the primary sign language followed in India, but there is little work on it because there is a lack of standardized datasets. Additionally, it involves two-hand gestures, which increase the difficulty of recognition due to occlusion of the hands. Most of existing tools for sign language learning use external sensors or complicated setups which are costly and hence difficult to get started with. In this work, commonly available standard hardware was used, with an aim to work with ISL and be able to perform rudimentary sign-language gesture recognition of image/video footage and benchmark few of the various suitable methods. Images are intended to be processed with various feature extraction techniques and pre-processing in order to enhance the efficacy of our implemented models. Our goal is to build a system that can provide robust hand sign-language gesture recognition for ISL. This can be of extensive help in public places, especially sign language that isn't often universally understood by the majority of people. Additionally, the work derives some conclusions on the usability of some of the methodologies and methods used in it.*

Key Words: ISL, Dataset, Skin Segmentation, Machine Learning, SVM, NBC, KNN, Feature Detection

1.INTRODUCTION

Sign language is used extensively by the hearing impaired, or even those that cannot speak, as usual language requires hearing as a way of starting to learn how to pronounce words. Although just communicating via written words in usual languages is still possible, sign language is present as a way of providing effective communication in these regards. Sign Languages have been created in many different regions, each with their own inflections and gestures, and the American Sign Language (ASL) and International Sign Language are popular styles of sign language. ASL is very popular for its simplicity and use of one-handed gestures which make it even more accessible. The Indian Sign Language(ISL) is a sign language that is used predominantly in the Indian subcontinent and shares a lot of commonalities with the British Sign Language, and is two-handed in its gestures. ISL in general uses gestures to communicate letters, numbers, as well as words as required.[1]

This work aims at identifying alphabets and numbers in Indian Sign languages from the corresponding gestures, and focuses on using lower-cost and easy to use camera hardware that is already highly prevalent in phones and laptops, instead of higher-end technologies involving the Kinect (a motion tracking camera) or the Leap Motion (Motion tracking gloves), which yields a solution that is easier to get started with and can be used for relatively less cost.

1.1 Motivation

Communication is a basic requirement , and Sign Language fills in this gap for people who cannot use traditional languages. Extensive work has been done regarding the American Sign Language (ASL), but more work can be done on ISL, which differs quite a bit from ASL. Here, two handed-gestures are common, and the gestures often vary based on the locality. This, along with the scarcity of datasets, has made it difficult to work on ISL. However, work is still being done slowly but surely towards getting solutions geared towards ISL. Most sign language projects aim to work through complex and high-end gear that can perform depth capture and motion capture. However, as these have not become common in every household, our work aims to work with common household-level cameras to aid in ISL Recognition.

2 Literature Survey

Evaluation of existing literature is an important aspect of any study. A thorough survey of available literature was carried out covering different domains under which the proposed study could be categorized. In most of the papers, the process for sign language recognition was performed using methods which included hardware support and deep learning techniques. A brief review of literature, in relevance to the proposed study, is summarized here.

In a study proposed by the JSPM's BSIOTR-Wagholi team[9] the authors have performed skin segmentation which was trained using UCI dataset and feature extraction using SIFT(Scale Inverse Feature Transformation), as a part of pre-processing. The classification was done using a linear kernel SVM giving 95% accuracy. A linear multi class SVM was also trained with alphabets data set with accuracy of 56% of single handed gesture and 60% for double hand gesture. In 2017 , a technical paper was published by Hore S. et al.

in Information Technology and Intelligent Transportation Systems [10] , in this publication the authors have proposed a novel approach for classification using three methods, Neural Networks, Genetic Algorithm, Evolutionary Algorithm and Particle Swarm optimization where the input weight vector to the Neural Network had been optimized gradually to achieve minimum error. The performance was compared to the Multi Layer Perceptron Feed Forward Network. The Neural Network - Particle Swarm Optimization outperformed the other approaches with 99.96% accuracy, 99.98% precision, 98.29% recall, 99.63% F-Measure and 0.9956% Kappa Statistic.

In a research conducted by Comilla University[11] the authors proposed a method for Bengali Sign Language recognition using Deep convolutional neural network. The method was built to recognize static hand signs of 37 letters of the Bengali alphabet. Fine tuning of the top layers of DCNN was done utilizing the learned features of the pretrained network. The model had achieved 96.33% recognition rate on training dataset and 84.68% on validation data set.

3. Design

3.1 Architecture

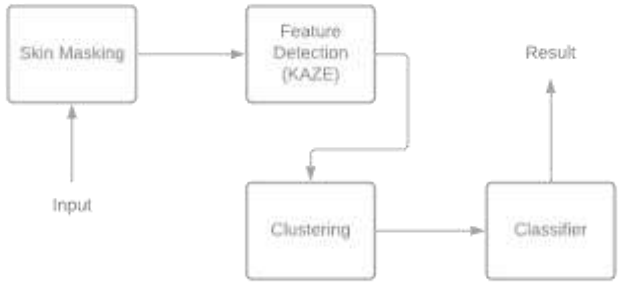

Fig -1:Architecture

Sign Language recognition is a comparatively complex problem and requires some preprocessing to be effective. We consider an architecture here that is designed as a pipeline:

1. Skin Segmentation: In this process all parts of the image other than the skin are removed. This removes unnecessary information, and can be done by various methods. Most popular methods rely on thresholding[3] HSV values and once resized, the image can be used accordingly.
2. Feature Detection: Feature detection is done using the KAZE feature detection algorithms that are bundled in *opencv-contrib*. This algorithm is a fast 2d multispace feature detection algorithm[4][5].
3. Clustering: Clustering allows us to build a `Visual Bag of Words` that represent the features detected from the algorithm. Here, the popular KMeans algorithm is used.
4. Classification: The last stage of the recognition is the classification. Here, three classification algorithms are used (KNNs, SVMs, NBCs), which are explained below. 5. Results: The results obtained are from the 3 models. Either one model can be chosen, or a majority vote may be considered.

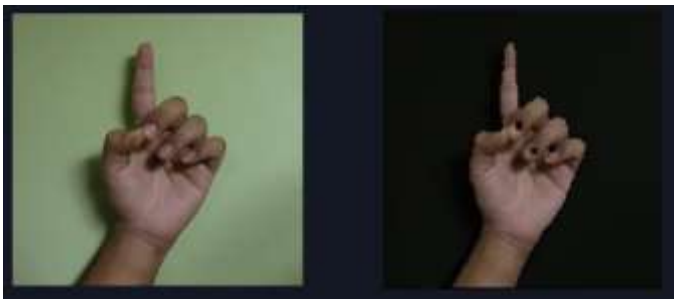

Fig -2: Skin Segmentation

For working with it as a solution, the system used for classification consists of API endpoints which provide the classification pipeline. Along with this, The system is divided into two service providers - a front-end API view layer and a back-end API provider.

3.2 Datasets Details

ISI datasets have been difficult to find, due to various problems related to the handedness, the difficulty to learn the language (As it varies from region to region slightly). Due to this, the work here involved looking and combining datasets that could be (feasibly) merged. Here, a final dataset was made with three components - A self made set that consisted of 100 images of ~25 classes each, publicly available datasets that provided around ~600 images of the 36 classes and some added commonalities to ASL, where two gestures that were the same across the languages were used.The above datasets were combined, preprocessed and shuffled. This obtained a set of wellmade and preprocessed data. Following this, the dataset was split into *24152/6030*; This forms a roughly 4:1 split which is sufficient for this medium-scale dataset. This dataset was then normalized, resizing it to a 128x128 image, as the different datasets had images of varying dimensions.

4. Implementation

A python-based application was created that could process the data and create models. For NBCs, the Bernoulli's,

Gaussian and Multinomial variants were used. Testing these, Bernoulli's method seemed to work the most favourably and was decided to be used. Similarly for KNNs, various values of *k* were used, and the best generalizing value was found and used for the model. Additionally, SVM was also used, and created models the fastest. Once the models were created, they could be served from an application to showcase the results.

4.1 Metrics

The process for working o n the proposed problem starts with searching for an existing data set corresponding to the project. After fiddling around with the data and checking through the current existing implementations, it needs to be normalized by using different, personal data inputs followed by segmentation and feature extraction and combined to be formed into a single data set. Once the data set is well defined and ready, the next step would be to set up the algorithms. When the algorithms have been finalized, a small UI was c reated to be able to manually test the results using images fed to the system. This entire flow has been summarized as a workflow diagram in Figure 3.

Fig -3:Workflow

All the models created were tested first with accuracy, and when the best variant was decided, additional metric data regarding the model were tested using the test set from the combined data set. The metrics tested for were accuracy, recall, and the f1-score, the last of which is simply a harmonic mean of the accuracy and recall, serving as a good indicator of how well the model is performing for the given test set.

4.2 Validation and testing

4.3 Environment

All of the models ran on the CPU and were conducted with a Ryzen 3200G and ran locally, with a prebuilt version of *opencv-contrib*. The models created run easily on most platforms, as there is no special GPU support required.

5. Results and Analysis

The models were created, stored, validated and used for inferencing; in the following sections the results and outputs are explained.

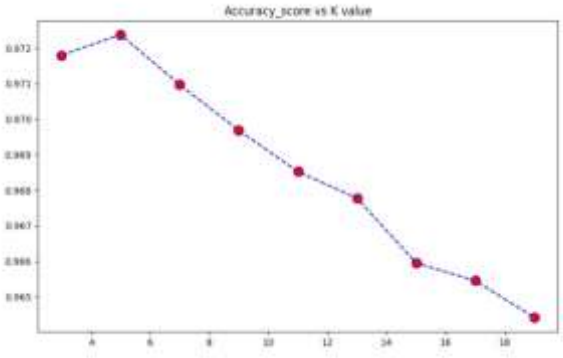

Fig -4:K vs accuracy (odd values only)

The above figure shows only odd values, as even values are often no t preferred to avoid a tie. Figure 6 indicates the results for k=7 , with an observed accuracy of 0.97, which performed nicely (the following value was chosen to aid in better generalization).

5.1. KNN

KNN algorithms are well known for classification because of their versatility. The results for the ISLR using kNN algorithm is shown as follows, Figure 9 represents the confusion matrix for kNN. The figure 10 displays a 'k vs Accuracy' graph indicating the decrease in accuracy with increase in the value k as expected. Odd values of K are used here, as they pose better accuracies, and theoretically, odd KNNs are easier to work with and present better accuracies as they can resolve ties easier. For making the model, 7KNN was chosen, as higher K values tend to generalize more.

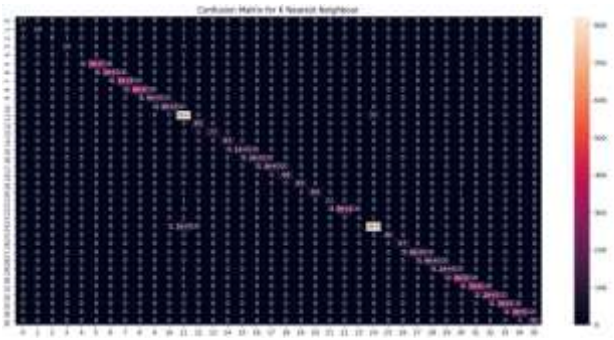

Fig -5:7KNN confusion matrix

accuracy			0.97	6014
macro avg	0.96	0.93	0.94	6014
weighted avg	0.97	0.97	0.97	6014

Fig -6: 7KNN accuracy and other metrics

5.2. SVM

SVMs were the easiest method to work with as they are relatively straightforward and require little work to get working. The performance was relatively good, but possibly requiring some improvement.

accuracy			0.98	6014
macro avg	0.96	0.96	0.96	6014
weighted avg	0.98	0.98	0.98	6014

Fig -7: SVM accuracy and other metrics

5.3. Naive Bayes Classification

Three types of Naive Bayes' algorithm: The Gaussian form,
Multinomial and Bernoulli's forms were considered. The In some cases, a misclassification may occur in one of the Bernoulli's form performed the most reliably (while cases, although the majority case seems to be able to manual testing) and was chosen. correct for this form of a mis-classification.

```
GNB
[25 25 25 ... 34 34 34]
0.8875956102427669
MNB
0.9459594280013303
BNB
0.9286664449617559
```

Fig -8: Accuracies of the NBC variants

accuracy			0.93	6014
macro avg	0.91	0.94	0.91	6014
weighted avg	0.93	0.93	0.93	6014

Fig -9: Metrics of the Bernoulli NBC

5.4. Overall Results

Here the models for the three given algorithms are used, and an additional column for “Majority” vote is used (KNN as fallback). The overall results are summed up as follows, indicating a few notable examples.

Type	Classification
SVM	L
7-KNN	1
G-NBC	L
Majority(FallBack KNN)	L

Fig -10: Sign for L

Type	Classification
SVM	9
7-KNN	1
G-NBC	9
Majority(FallBack KNN)	9

Fig -11: Sign for 9, KNN classifying as 1 An irrelevant image not containing a sign, also is classified correctly as 'None'.

Type	Classification
SVM	None
7-KNN	None
G-NBC	None
Majority(FallBack KNN)	None

Fig -12: A hand drawing classified as not a hand sign

6. CONCLUSIONS

The motivation behind the work was a comparative study on the various models, and present a better picture of what would work better, and under what circumstances. Along with this, helping expand the set of available ISL gestures, the work here should also help future work in this scope.

The results of our models, and statistics were shown in previous sections and it came as a conclusion that the SVM was the best performing model. Continuing this ahead, the UI solution that was created had a majority vote system that would pick the solution that was picked twice (or fall back to KNN if required) between the 3 models. The resultant solution was a well-scalable classifier that could work with images and respond and identify parts of ISL. Covering some of the shortcomings, the solution could provide a solution for recognition of Indian Sign Language.

6.1 Limitations and Future Work

Although given the good performance of the metrics, there are a lot of limitations that exist in the solution. Firstly, the dataset is large, but even then, for computer-vision projects, it is relatively small; more data could be helpful. This is a difficult limitation to get around as ISL is varied, and is two-handled. Further, the solution was limited to only the character and number subset in the ISL. Following this, there are various subsets of the ISL that are ambiguous, and require context to clarify their meaning. Additionally, the model does not capture temporal inflections in ISL, and does not perform any hand detection; this must be done by the user, or by another solution added to the pipeline. Additionally, other methods such as Linear Discriminant Analysis (LDA) or even Convolutional Neural Networks (CNN) may prove to be useful.

REFERENCES

[1] JOYEETA SINGH, K. D. Indian sign language recognition using eigen value weighted euclidean distance based classification technique. International Journal of Advanced Computer Science and Applications 4, 2 (2013). NEHA V. TAVARI, P. A. V. D. Indian sign language recognition based on histograms of oriented gradient. International Journal of Computer Science and Information Technologies 5, 3 (2014), 3657–3660.

[2] Aizerman, Mark A.; Braverman, Emmanuel M. & Rozonoer, Lev I. (1964). "Theoretical foundations of the potential function method in pattern recognition learning". Automation and Remote Control. 25: 821–837.

[3] N. Dwina, F. Arnia and K. Munadi, "Skin segmentation based on improved thresholding method," 2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON), 2018, pp. 95-99, doi: 10.1109/ECTI-NCON.2018.8378289.

[4] Fernández Alcantarilla, Pablo & Bartoli, Adrien & Davison, Andrew. (2012). KAZE Features. 10.1007/978-3642-33783-3_16.

[5] Tareen, Shaharyar Ahmed Khan & Saleem, Zahra. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. 10.1109/ICOMET.2018.8346440.

[6] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, Weiping Li arXiv:1801.10111 [cs.CV]

[7] Sirshendu Hore, Sankhadeep Chatterjee, V. Santhi, Nilanjan Dey,Amira S. Ashour, Valentina Emilia Balas and Fuqian Shi, "Indian Sign Language Recognition Using Optimized Neural Networks" , Springer International Publishing Switzerland 2017 V.E.Balas et al. (eds.), Information Technology and Intelligent Transportation Systems, Advances in Intelligent Systems and Computing 455,DOI 10.1007/978-3-319-38771-0-54.

[8] M.A Hossen, Arun Govindaiah, Sadia Sultana and Alauddin Bhuiyan , "Bengali Sign Language Recognition Using Deep Convolutional Neural Network" ,2018 Joint 7th International Conference on Informatics, Electronics - Vision (ICIEV) and 2018 2ndInternational Conference on Imaging, Vision & Pattern Recognition (icIVPR).

[9] Prof. Radha S. Shirbhate, Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar ,Ms. Mayuri A. Khandge , “Sign language Recognition Using Machine Learning Algorithm” , International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 , p-ISSN: 2395-0072.

[10] Hore S. et al. (2017) Indian Sign Language Recognition Using Optimized Neural Networks. In: Balas V., Jain L., Zhao X. (eds) Information Technology and Intelligent Transportation Systems. Advances in Intelligent Systems and Computing, vol 455. Springer, Cham.https://doi.org/10.1007/978-3-319-38771-0_54

[11] M. A. Hossen, A. Govindaiah, S. Sultana and A. Bhuiyan, "Bengali Sign Language Recognition Using Deep Convolutional Neural Network," 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2018, pp. 369-373, doi: 10.1109/ICIEV.2018.8640962.