RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi, Accredited By NAAC, Bengaluru And NBA, New Delhi

Go, change the world

Sign Language Recognition Systems

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB (18CS62)

MINOR PROJECT REPORT

Submitted by

ATREYA BAIN	1RV18CS030
BIRAJDAR SHIWAM SANJAYKUMAR	1RV18CS044

in partial fulfillment for the requirement of 6th Semester Artificial Intelligence and Machine Learning(18CS62)

Under the guidance of

Dr. Manonmani, Associate Professor
Dept. of Computer Science, R. V. College of Engineering

Academic Year 2020-21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Atreya Bain and Birajdar Shiwam Sanjaykumar, students of eighth semester B.E., department of CSE, RV College of Engineering, Bengaluru, hereby declare that the minor project titled **'Sign Language Recognition Systems'** has been carried out by us and submitted in partial fulfilment for the award of degree of Bachelor of Engineering in Computer Science and Engineering during the year 2020-21.

Further we declare that the content of the report has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru
Date:

Name	**Signature**
1. ATREYA BAIN (1RV18CS030)	
2. BIRAJDAR SHIWAM SANJAYKUMAR (1RV18CS044)	

RV College of Engineering

Bengaluru 560 059

Autonomous Institution Affiliated to VTU, Belagavi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work titled ‘Sign Language Recognition Systems’ is carried out by Atreya Bain(1RV18CS030), Birajdar Shiwam Sanjaykumar(1RV18CS044), who are bonafide students of RV College of Engineering®, Bengaluru, in partial fulfillment of the curriculum requirement of 6th Semester Artificial Intelligence and Machine Learning during the academic year **2020-21**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in all respect as prescribed by the institution.

Signature of faculty in-charge

Head of the Department
Dept. of CSE, RVCE

External Examination

Name of Examiners **Signature with date**

1.

2.

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work and, I would like to take this opportunity to thank them all.

I deeply express my sincere gratitude to my guide **Dr. Manonmani**, Associate Professor, Department of Computer Science and Engineering, RVCE, Bengaluru, for his able guidance, regular source of encouragement and assistance throughout this project.

I would like to thank **Dr. Ramakanth Kumar P**, Head of Department, Computer Science and Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

First and foremost I would like to thank **Dr. Subramanya K. N**, Principal, R.V.C.E, Bengaluru, for his moral support towards completing my project work.

I thank my Parents, and all the Faculty members of Department of Computer Science and Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

Abstract

Sign language is a good visual communication aid for those with auditory disablities. This language is also prevalent for those with speech impairment. However, the general populus 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 the existing tools for sign language learning use external sensors which are costly. However, for our work, we plan to use standard hardware that is very commonly available and cheap to use for our setup.

Our 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 isn't often universally understood by the majority of people.

This can also be useful for captioning video segments involving ISL.

Contents

1 Introduction 1

1.1 Project Domain . . . 1

1.2 Issues and Challenges . . . 2

1.3 Need for AI-based solutions . . . 2

1.4 Problem Statement . . . 2

1.5 Project objectives . . . 3

2 Literature Study 4

2.1 SLR using Machine Learning Algorithms . . . 4

2.2 ISLR using optimized neural networks . . . 4

2.3 Bengali Sign Language Recognition Using Deep Convolutional Neural Network 5

2.4 Independent Bayesian classifier combination based SLR using facial expression 6

2.5 Visual Alignment Constraint for Continuous Sign Language Recognition . . . 6

2.6 Real-time slr using a consumer depth camera . . . 7

2.7 Indian Sign Language character recognition using neural networks . . . 7

2.8 Video-Based Sign Language Recognition without Temporal Segmentation . . . 8

2.9 SLT :Joint End-to-end Sign Language Recognition and Translation . . . 8

2.10 Isolated Sign Language Recognition using Hidden Markov Models . . . 9

2.11 Real Time ASL recognition from video using Hidden Markov Model . . . 10

2.12 Australian sign language recognition . . . 11

3 Design 12

3.1 Architecture . . . 12

3.1.1 System Architecture . . . 12

3.2 Methodology . . . 13

3.3 Dataset details . . . 14

3.4 ML Techniques used . . . 14

3.4.1 Classification Techniques . . . 14

3.5 Hardware and Software Requirements . . . 17

3.5.1 Software Requirements . . . 17

3.5.2 Hardware Requirements . . . 18

4 Implementation 18

4.1 Languages/Tools/APIs used . . . 18

4.2 Use cases . . . 19

4.3 Workflow Diagrams . . . 19

4.4 Data Preprocessing . . . 20

4.5 Validation Methodology . . . 21

5 Result and Analysis **21**

5.1 KNN . . . 21

5.2 SVM . . . 22

5.3 NBC . . . 23

5.4 Overall Results . . . 24

6 Conclusion and Future Enhancement **25**

6.1 Novelty in the proposed solution . . . 25

6.2 Limitations . . . 25

6.3 Future Enhancements . . . 25

6.4 Conclusion . . . 26

List of Figures

1 Software Architecture for Classification 12
2 System Architecture for Solution 13
3 k-NN infographic 15
4 SVM for 2 features 16
5 Kernel Machines - Making it ‘linear‘ 16
6 Naive Bayes Classifier infographic 17
7 Workflow Diagram 19
8 Skin Segmentation 20
9 Confusion matrix for KNN 21
10 k vs Accuracy 22
11 7KNN Statistics 22
12 SVM confusion matrix 22
13 SVM Statistics 23
14 Bernoulli NBC confusion matrix 23
15 Accuracies of various NBC forms 23
16 NBC Statistics 23
17 None and 9 24
18 2 and L 24
19 2 29
20 L 29
21 None 30
22 L 30

List of Tables

1 Introduction

The sign language is used widely by people who are deaf-dumb these are used as a medium for communication. Sign language is composed of various gestures formed by different placements of hand, its movements,orientations as well as the facial expressions. There are around 466 million people worldwide with hearing loss and 34 million of these are children. 'Deaf' people have very little or no hearing ability. They use sign language for communication. People use different sign languages in different parts of the world. Compared to spoken languages they are very less in number. India has its own sign language by the name Indian Sign Language (ISL). In developing countries there are only very few schools for deaf students. Unemployment rate among adults with hearing loss are very high in developing countries. Data states that among deaf population in India, which is about 1 percent of total population, literacy rate and number of children attending school is very less.It goes on to state that official recognition of sign languages, increasing the availability of interpreters and providing transcription in sign languages greatly improve accessibility. Signs in sign languages are the equivalent of words in spoken languages Signed languages appear to favor.

This project aims at identifying alphabets and numbers in Indian Sign Language from the corresponding gestures. Gesture recognition and sign language recognition has been a well researched topic for American Sign Language(ASL), but few research works have been published regarding Indian Sign Language(ISL). This, is because the ISL is a two-handed sign language, and there are sometimes dialects that are needed to be preserved. 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.

1.1 Project Domain

The Project falls under the following spectrum :

- The proposed problem lies in the domain of Machine learning for classification and pattern recognition
- Machine learning is method of data analysis which builds an analytical model . It's based on the principle that systems can learn from previous data and identify patterns and make decisions.
- The proposed problem falls under classification and identification of patterns in a given data set of sign language gestures
- Methods such as Support Vector Machines can be used for classification problems.

- Neural Networks are also used for classification and pattern matching. They are composed of parallel subunits called neurons which have the ability to change weights with every iteration and adjust to a set of unique weights which give optimal results for the proposed problem

1.2 Issues and Challenges

The issues and challenges faced could be summarized as follows:

- The Indian Sign Language uses many gestures with two hands involved as opposed to just one in American and other International Sign Languages. Hence feature extraction would be a challenge
- Considering the fact that we're dealing with Indian Sign Language recognition, the skin color also comes into the picture making the current existing models for ASL inefficient when it comes to pre processing.
- Acquiring a proper data set is also a major challenge in this case as it involves different orientations , shapes and movements of hands.
- Choosing the proper classification algorithm is also a major challenge, we try to get a results comparison for the three major ones
- For gaining more diverse recognition for gestures we need to train larger data sets going up to 4.5 GBs and more, hence computing power would be an issue

1.3 Need for AI-based solutions

The need for an AI solution is because :

- Hand sign recognition is a notoriously difficult problem for computers to solve. It involves gesture recognition, computer vision and many such sub-problems, which are innately difficult.
- Additionally, some parts of sign language are also time-sensitive, making this problem multi-dimensional in nature.
- AI has often been used to tackle such problems in recent problems, and shows great promise in this field.

1.4 Problem Statement

To create and develop a machine learning model to recognize and Interpret Sign language using classification algorithms

1.5 Project objectives

Project Objectives include:

- To preprocess incoming image/video footage with various feature extraction techniques and pre-processing in order to enhance the efficacy of our implemented models.
- To work with existing datasets to preprocess and collect them for usage in the creation of models.
- To source incoming video and image footage from low-cost and already present camera hardware.
- To Showcase various models that can recognize a subset of the Indian Sign Language
- To be able to perform rudimentary sign-language gesture recognition of image/video footage
- Benchmark few of the various suitable methods, and show their accuracy and precision
- To Present a model that can (comparatively speaking) reliably recognize ISL so as to Be able to convert ISL to text

2 Literature Study

Literature Study of various papers were carried out in the domain of Indian, American and International Sign Language Recognition techniques. The papers cover a large spectrum of methodologies and classification techniques. The detailed review of each paper is carried out in forthcoming subsections.

2.1 SLR using Machine Learning Algorithms

Sign Language Recognition using Machine Learning algorithms was released by Prof. Radha S. Shirbhate, Mr. Vedant , D. Shinde, Ms. Sanam, A. Metkari, Ms. Pooja , U. Borkar, Ms. Mayuri and A. Khandge in IRJET Volume: 07 Issue: 03 — Mar 2020. [5] The objective of the paper was as follows , Extensive work has been done on American sign language recognition but Indian sign language differs significantly from American sign Language (ASL). ISL uses two hands for communicating whereas ASL is one-handed. Using both hands often leads to security of features due to overlapping of hands. In addition to this, lack of datasets along with variance in sign language with locality has resulted in restrained efforts in ISL gesture detection. The project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using Indian sign language.
The Results obtained by the authors were stated as , The data set divided into two groups, one used for training and other for testing. The training set consists of 70 percent of the aggregate data and remaining 30 percent are used as testing. We also perform experiments on same (30percent or 70 percent) dataset which is training as well as testing for KNN classifier. The results on these experiments have a 100 percent accuracy rate. This means, if the user who is supposed to use this project has already contributed to the dataset earlier, the system will guarantee 100 percent recognition rate.
The limitations of the paper can be put forth as follows , The system can be useful for static ISL numeral signs only.The ISL recognizer system cannot be considered as a complete system, as for complete recognition of sign language, they have to include ISL alphabets, words and sentences. These signs can be included in future. Also other feature extraction algorithms like Wavelet transform, Invariant moments, Shape lets descriptors and other existing methods can be included in conducting experiments for improvement in the results. Other classifiers like multi class Support Vector Machine (SVM), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) or a combination of these classifiers can be included in conducting experiments to improve the recognition rate.

2.2 ISLR using optimized neural networks

Indian Sign Language Recognition Using Optimized Neural Networks was written by Sirshendu Hore, Sankhadeep Chatterjee, V. Santhi, Nilanjan Dey,Amira S. Ashour, Valentina Emilia

Balas and Fuqian Shi and published in 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.[6]

The objectives of the paper were as follows , A four step method to recognize alphabets of ISL has been proposed in. In a statistical approach to classify and recognize dynamic gestures of ISL in real time was proposed. Orientation of histogram has been used as a key feature for classification. Edge orientation of the sequence of dynamic ISL gestures has been calculated by employing a simple algorithm. K-nearest neighbor and Euclidean distance have been used for classification.The authors have reported accuracy range of 51.35 to 100 and about 64.42 to 100 percent.

The results displayed were, unlike classifier-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. .It also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection. The limitations were Localization errors account for more of YOLO's errors than all other sources combined.It struggles with small objects that appear in groups.

2.3 Bengali Sign Language Recognition Using Deep Convolutional Neural Network

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

The objectives covered in the paper were , a new method for Bengali Sign Language Recognition using Deep Convolutional Neural Networks(DCNN).The method is built to recognize static hand signs of 37 letters of the Bengali alphabet. Conducting tests on three sets of 37 signs (we have used 31 sets of images for 37 different signs) with total 1147 images, with varying the precision of feature points taken on each test. With the use of deep convolutional neural network and utilizing the learned features from a pre-trained network and fine-tuning the top layers of this network.

In order to test their method, they created two extra sets of 37 signs with the non-skin colored background. They have achieved, with their proposed recognition system, the following results - validation loss of 0.3523 (categorical cross-entropy) and validation accuracy of 84.68 percent . This accuracy is very high considering the small size of the data set.Larger and diverse data set required for efficient output Advanced Neural network architecture is needed to get almost perfect results, here pre-trained dataset with small amount of variations was used.

2.4 Independent Bayesian classifier combination based SLR using facial expression

Independent Bayesian classifier combination based sign language recognition using facial expression was written by Pradeep Kumar, Partha Pratim Roy and Debi Prosad Dogra in Elsevier Information Sciences Article history:Received 7 September 2016,Revised 3 October 2017,Accepted 23 October 2017,Available online 27 October 2017.[8]
The objective of the paper was to propose a model, DenseNet with strong feature extraction ability and high parameter efficiency to achieve object detection combined with Region Proposal Network(RPN). Hand movements and facial expressions are captured simultaneously by using Leap motion and Kinect, respectively. Next, a Bayesian classifier combination model IBCC is applied to combine the decision from the two modalities which results in better recognition performance than uni-modal and feature combination based recognition approaches. They have achieved 96.05 percent and 94.27 percent recognition rate in single and double hand gestures, respectively.
The limitations are more robust features can be extracted from both hand and facial data. Moreover, other classification model and classifier combination approaches could be used for improving recognition results.The results for double handed gestures are lower in comparison to single handed gestures. It is because of the noise occurred while signing using double hands.

2.5 Visual Alignment Constraint for Continuous Sign Language Recognition

Visual Alignment Constraint for Continuous Sign Language Recognition was written by Yuecong Min, Aiming Hao, Xiujuan Chai and Xilin Chen published in arXiv:2104.02330v1 [cs.CV] 6 Apr 2021.[9] In objectives of the project They revisit the overfitting problem in CSLR and attribute it to the insufficient training of the feature extractor with limited training data. They propose a visual alignment constrain to make the network end-to-end trainable by enhancing the feature extractor and aligning visual and contextual features. They present two metrics to evaluate the contributions of the feature extractor and the alignment model, which verifies the effectiveness of the proposed method
In this work, analytical experiments show that original end-to-end training leads to insufficient training for the feature extractor due to the overfitting problem. To solve this problem, we propose the visual alignment constraint to make CSLR networks end-to-end trainable by enforcing the feature extractor to make predictions with more alignment supervision. Two new metrics are presented to evaluate the different behaviors of the feature extractor and the alignment model. Experimental results verify the proposed VAC successfully narrows the gap between two classifiers. The proposed metrics and relevant analysis provide a new perspective on the relationship between visual and alignment models .Its easy to overfit, changes could be

made to better understand visual information.

2.6 Real-time slr using a consumer depth camera

Real-time sign language recognition using a consumer depth camera was written by Alina Kuznetsova, Laura Leal-Taixe and Bodo Rosenhahn from Institute fuer Informationsverarbeitung, Leibniz University Hannover Appelstr. 9A, Hannover, 30167, Germany.[10] The objective of the paper can be explained in two parts ,first they propose to use rotation-, translation- and scaling- invariant image features to reduce intra-class variation. Such features exist for 3D point clouds, which can be derived from the depth data, delivered by a depth sensor. These features can be separated into two categories — local and global features. Secondly, they propose to use multi-layered random forest (MLRF) for classification.The intuition behind this approach is to find groups of similar feature descriptors and for each group train a separate classifier that can better distinguish the classes withing this group.

In the performance of the shape classification for the dataset is presented. In the l-o-o experiment, the achieved accuracy is 85 vs. 57 percent of their method and in the h-h experiment, the reported performance is 97.8 percent . However, the reported training time is around 4000 s per tree on a quad core PC. For their method, the training time for an unoptimized MATLAB version of the MLRF is less then one thousand of seconds on one core. In one-subject experiment, Their method shows the accuracy of 97.4 percent . Therefore, in one-subject experiment their method has the same performance as the method in while the training times allows to re-calibrate the system for a new subject very fast.

The authors evaluate the performance of their method depending on the key parameters and compare it with the state-of-the-art methods. Their method demonstrates low training time,low memory usage and high accuracy in one subject tests,which make it applicable in a scenario, when fast system re-calibration is acceptable.

2.7 Indian Sign Language character recognition using neural networks

Indian Sign Language character recognition using neural networks was written by Saipreethy.M.S, Valliammai . V and Padmavathi . S from department of IT , Amrita vishwa vidyalaya.[11] The methodology followed was , the alphabets in Indian Sign Language , uses both the hands which differentiate it from American Sign Language. Indian Sign Language characters are almost similar to the characters themselves. The alphabets could be characterized by the angle between the fingers, which finger is open, how much the particular finger is open and how many fingers are open. The input video is converted into frames and HSI color model based segmentation is applied on each frame.

After segmentation, in feature extraction, certain features of the hand like centroid of the hand helps identify features like posture of the fingers and angle between them. After the phase of extraction, is the recognition phase where the extracted features are fed into the neural network to recognize the particular character.
At the end they concluded that proposed back propagation neural network has the highest accuracy with 99 percent , precision 89.47 percent , recall 89.78 percent and specificity 90.54 percent.

2.8 Video-Based Sign Language Recognition without Temporal Segmentation

Video-Based Sign Language Recognition without Temporal Segmentation written by Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li and Weiping Li published in The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).[12]
The objectives of this paper was to show new two-stream 3D CNN for the generation of global- local video feature representations, a new LS-HAN framework for continuous SLR without requiring temporal segmentation; Joint optimization of relevance and recognition loss in the proposed LS-HAN framework; Compilation of the largest (as of September 2017) open source Modern Chinese Sign Language (CSL) dataset for continuous SLR with sentence-level annotations.
In this paper, the LS-HAN framework is proposed for continuous SLR is proposed, which eliminated both the error prone temporal segmentation and the sentences synthesis in post-processing steps. For video representation, a two stream 3D CNN generates highly informative global-local features, with one stream focused on global motion information and the other on local gesture representations.A Latent Space is subsequently introduced via an optimization of labeled video-sentence distance metrics. This latent space captures the temporal structures between signing videos and annotated sentences by aligning frames to words. Our future work could involve the extension of the LS-HAN to longer compound sentences and real-time translation tasks.

2.9 SLT :Joint End-to-end Sign Language Recognition and Translation

Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation is written by Necati Cihan Camgoz ” , Oscar Kollerq, Simon Hadfield and Richard Bowden. Published on IEEE XPLORE.[13] The objectives of this paper can be summarized as A novel multi-task formalization of CSLR and SLT which exploits the supervision power of glosses, without limiting the translation to spoken language. The first successful application of transformers for CSLR and SLT which achieves state-of-the-art results in both recognition and translation accuracy, vastly outperforming all comparable previous approaches.A broad range of new baseline results to guide future research in this field.

They first go over the implementation details and introduce the evaluation metrics we will be using to measure the performance of our models. They start our experiments by applying transformer networks to the text-to-text based SLT tasks, namely Gloss2Text, Sign2Gloss2Text,Sign2Gloss→Gloss2Text and report improved performance over using Recurrent Neural Network (RNN) based models.They share our Sign2Gloss experiments, in which we explore the effects of different types of spatial embeddings and network structures on the performance of CSLR.

Sign language recognition and understanding is an essential part of the sign language translation task. Previous translation approaches relied heavily on recognition as the initial step of their system. In this paper they proposed Sign Language Transformers, a novel transformer based architecture to jointly learn sign language recognition and translation in an end-to-end manner. They utilized CTC loss to inject gloss level supervision into the transformer encoder,training it to do sign language recognition while learning meaningful representations for the end goal of sign language translation, without having an explicit gloss representation as an information bottleneck.

2.10 Isolated Sign Language Recognition using Hidden Markov Models

Isolated Sign Language Recognition using Hidden Markov Models written by Kirsti Grobel and Marcel1 Assan for IEEE Cybernetics conference.[14] This paper is concemed with the video-based recognition of isolated signs. Concentrating on the manual parameters of sign language, the system aims for the signer dependent recognirion of 262 different signs. For Hidden Markov Modelling a sign is considered a doubly stochastic process, represented by an unobservable state sequence. The observations emitted by rhe states are regarded as feature vectors, that are extracted from video frames. The system achieves recognition rates up to 94 percent.

The property of Hidden Markov Models (HMMs) to compensate time and amplitude variances of signals has been proven for speech and character recognition. This property makes HMMs appear an ideal approach for sign language recognition.Like speech, sign language can be considered as a non deterministic time signal, not as a word or phoneme sequence but as a sequence of signs. Unlike speech recognition, where the smallest unit is the phoneme, linguists have not agreed on first proposals on sub-units for signs. Thus we model each sign with one HMM. For both, training and recognition, feature vectors must be extracted from each video frame and then inputted to the HMM. The presented system for the recognition of isolated signs only regards the manual parameters, as non-manual parameters often have grammatical functions or emphasise emotions. The system is capable of recognising 262 different signs of the Sign Language of the Netherlands. The aim is signer dependent recognition, i.e. the same person trains and tests the system.

They present a video-based system for the recognition of isolated signs with little intrusion on the signer. The system is equipped with a single camera in order to minimise the hardware requirements. We described how the HMM theory adapts to sign language recognition and presented details for the training and recognition procedures. With a feature vector of relatively simple features the results prove that even 262 different signs from two signers can be discriminated with a high probability. As the results are very promising, the next steps will be a further expansion of the vocabulary. A challenging future task will be the transition to connected sign recognition, allowing more comfortable human-machine interaction.

2.11 Real Time ASL recognition from video using Hidden Markov Model

Real Time ASL recognition from video using Hidden Markov Model authored by THAD STARNER AND ALEX PENTLAND from MIT Boston[15].

They describe an extensible system which uses one color camera to track hands in real time and interprets American Sign Language (ASL) using Hidden Markov Models (HMM's). The hand tracking stage of the system does not attempt a fine description of hand shape; studies of human sign readers have shown that such detailed information is not necessary for humans to interpret sign language. Instead, the tracking process produces only a coarse description of hand shape, orientation, and trajectory. The hands are tracked by their color: in the first experiment via solidly colored gloves and in the second, via their natural skin tone. In both cases the resultant shape, orientation, and trajectory information is input to a HMM for recognition of the signed words. Hidden Markov models have intrinsic properties which make them very attractive for sign language recognition.

They have shown an unencumbered, vision-based method of recognizing American Sign Language (ASL). Through use of hidden Markov models, low error rates were achieved on both the training set and an independent test set without invoking complex models of the hands.With a larger training set and context modeling, lower error rates are expected and generalization to a freer, user independent ASL recognition system should be attainable. The future enhancements of this paper can be summarized as :

- Measure hand position relative to each respective shoulder or a fixed point on the body.

- Add finger and palm tracking information. This may be as simple as counting how many fingers are visible along the contour of the hand, and whether the palm is facing up or down.

- U se a two camera vision system to help disambiguate the hands in 2D and/or track the hands in 3D.

- Collect appropriate domain or task-oriented data and perform context modeling both on the trisine level as well as the grammar/phrase level.

- Integrate explicit face tracking and facial gestures into the feature set.

2.12 Australian sign language recognition

Australian sign language recognition authored by Eun-Jung Holden , Gareth Lee and Robyn Owens for Machine Vision and Applications (2005) 16(5): 312–320[16]. This paper presents an automatic Australian sign language (Auslan) recognition system, which tracks multiple target objects (the face and hands) throughout an image sequence and extracts features for the recognition ofsign phrases.

Tracking is performed using correspondences of simple geometrical features between the target objects within the current and the previous frames. In signing, the face and a hand of a signer often overlap, thus the system needs to segment these for the purpose of feature extraction. The system deals with the occlusion of the face and a hand by detecting the contour of the foreground moving object using a combination of motion cues and the snake algorithm. To represent signs, features that are invariant to scaling, 2D rotations and signing speed are used for recognition. The features represent the relative geometrical positioning and shapes of the target objects, as well as their directions of motion. These are used to recognise Auslan phrases using Hidden Markov Models. Experiments were conducted using 163 test sign phrases with varying grammatical formations. Using a known grammar, the system achieved over 97 percent recognition rate on a sentence level and 99 percent success rate at a word level.

The proposed system has the following limitations that will be considered for future developments. The tracking algorithm uses a combination of skin colour detection and a simple geometric correspondence algorithm. If skin coloured objects appear in complex backgrounds or the signer's clothing, with the similar shape and location of the target objects, the tracking may fail to determine the correspondence. The use of a prediction algorithm using spatio-temporal velocity may be difficult as a hand changes its direction suddenly causing the discontinued velocity. To deal with complex backgrounds, the system may require the tracking of elbows or other physiological landmarks. The segmentation algorithm has been tested with moving foreground objects, but does not deal with the background object changing shape.

3 Design

3.1 Architecture

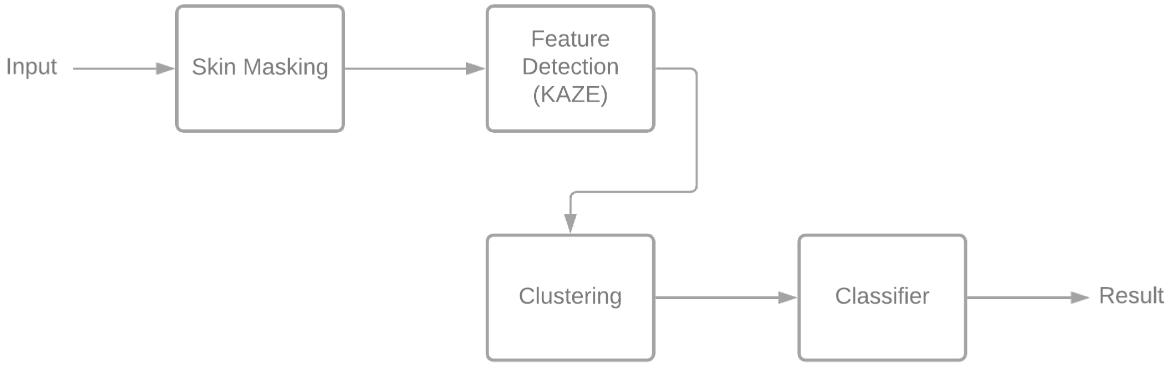


Figure 1: Software Architecture for Classification

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 Masking: Skin Masking is a process of removing all parts of the image other than the skin. This removes unnecessary information, and can be done by various methods. Most popular methods rely on thresholding[2] 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.

3. Clustering: Clustering allows us to build a ‘Visual Bag of Words‘ that represent the features detected from the algorithm.

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.

3.1.1 System Architecture

The system used for classification consists of an 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.

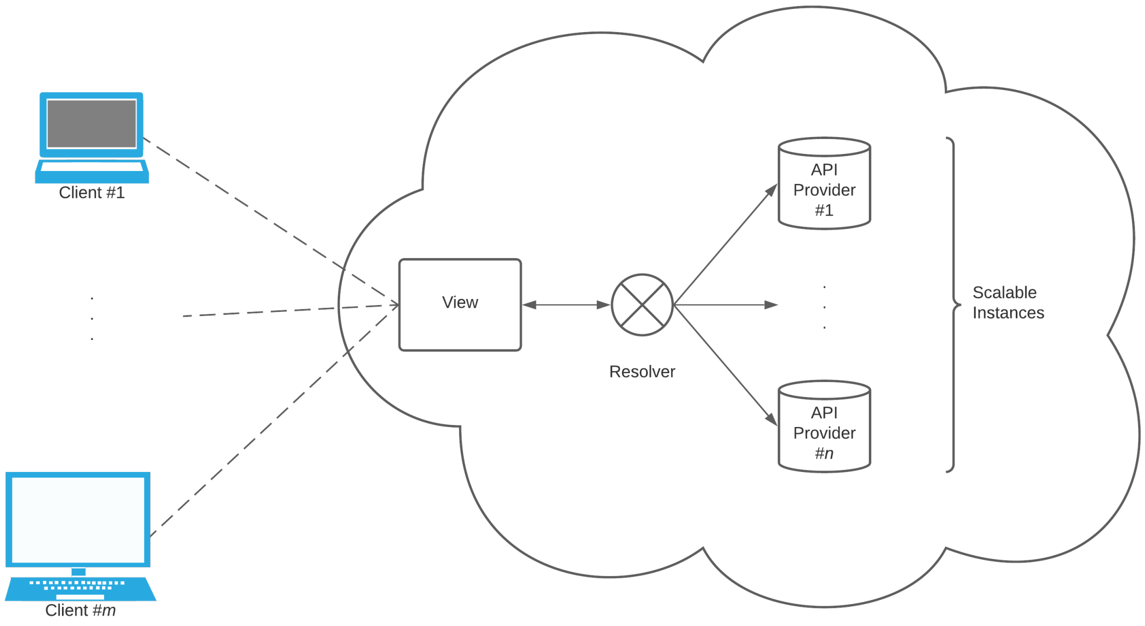


Figure 2: System Architecture for Solution

3.2 Methodology

The following procedures have been followed :

1. The process of sign language interpretation can be divided into three computations :
   - Skin Segmentation
   - Feature Recognition
   - Classification
2. Skin segmentation aims at partitioning the skin present in the image based on various features eg. color, shape or texture. This preprocessing step helps remove unnecessary detail in the image to help improve efficacy of our system.
3. The feature recognition part of the process aims at recognizing the prominent and necessary features of the hand such as palm and finger tips. This would help us in narrowing down the necessary features which are to be worked on during the later stages. It is most popularly implemented using open source modules such as OpenCV.
4. Classification is the crux of our project. We propose to test classification algorithms such as Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Naive Bayes Classifier (NBC) and document their working and efficacy at Sign Language interpretation.
5. The system must be fed with video input that has been appropriately pre-processed. The aim is to ensure that the hardware required is minimal. The output is intended to

be the text that has been translated/interpreted. The text may be interpreted separately or be embedded as captions.

3.3 Dataset details

A combination of datasets were used to obtain the final dataset. Three components datasets were used where:

- Self-Made Dataset: ˜100 images of 25 classes each. This excluded some tough to capture gestures.
- Available popular Datasets: ˜600 images of 36 classes.
- ASL commonalities: Gestures that are common to ASL and ISL can be used. ˜50 images for the 2 common gestures were used.

The above datasets were combined, preprocessed and shuffled. This obtained a set of well-made 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.

3.4 ML Techniques used

3.4.1 Classification Techniques

Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.The classification methodologies used in the proposed project are K-Nearest neighbour(KNN), Support vector machines(SVM) and Naive-bayes(NBCs).

1) k-Nearest Neighbour algorithm

The k-nearest neighbors (kNN) algorithm is a supervised machine learning algorithm that can be used to solve both classification and regression problems. kNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label. This algorithm is easier to implement and is known for its versatility.

The initial step for the workings of a kNN algorithm are to find the distance between the selected query and the other queries in the dataset. The distance is calculated using the euclidean formula or the Manhattan metrics, the most popular being Euclidean distance. After calculating the distance , one can decide how many closest neighbour one wants to choose for the classification of that query. After selection of the 'k' nearest queries, the values of the queries are averaged to find the class of that particular set of queries and the newest query is assigned to it.

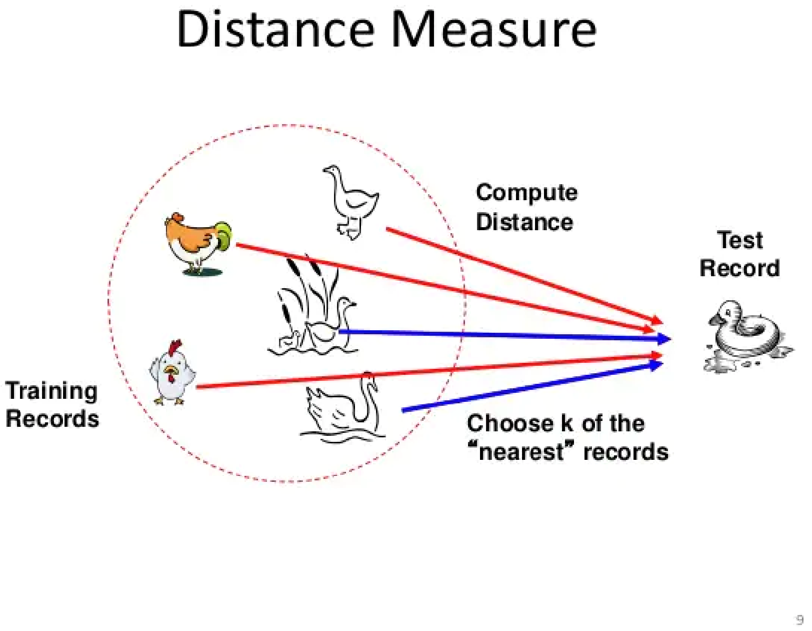


Figure 3: k-NN infographic

The KNN algorithm is very useful as it often defers actual computations to the classification time, although for large datasets searching and computing the distances may quickly become expensive.

2) Support Vector Machines

Support Vector machines are a popular choice for supervised Machine Learning. They are used very often for classification and may be extended for regression as well, and work by adjusting a hyperplane according to the data in an n-dimensional space. When it comes to SVM, there are two popular SVM forms: hard margin SVM and soft margin SVMs. Hard Margin SVMs are preferred in the case where it is known that the training data is linearly separable and soft margins are an extension of the former using a hinge loss function, that allow the use of non-linearly separable data as well.

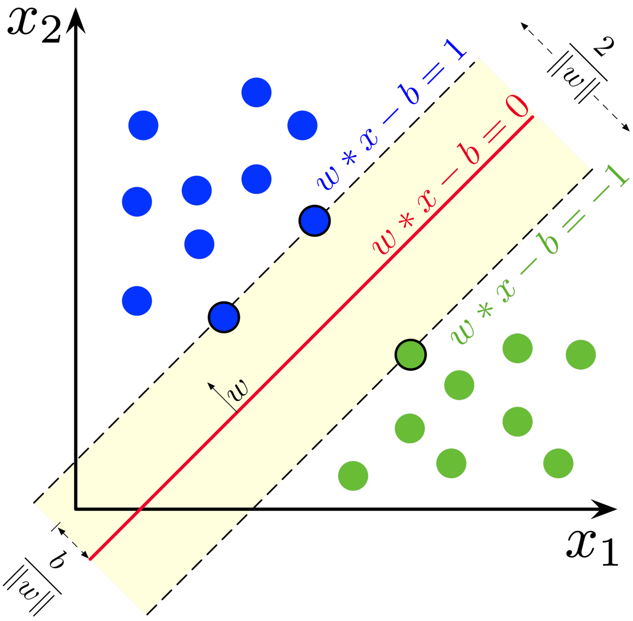


Figure 4: Support Vector machines - Maximizing the hyperplane margins between the support vectors

SVMs are not limited to linear classification, but may also be extended to non-linear classification by the application of kernel function (known as a 'kernel trick' informally)[1].

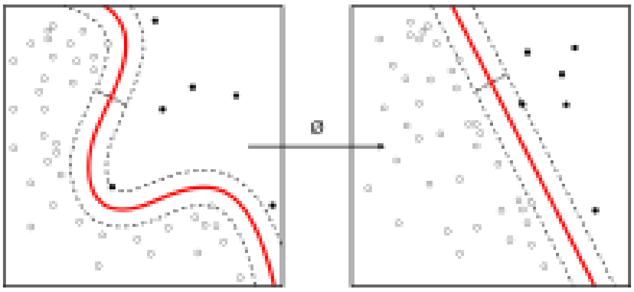


Figure 5: Kernel Machines - Making it 'linear'

This version of SVMs classify in a higher dimensional feature space to accomplish the 'linear' fitting. It is important to note that working in a higher-dimensional space does increase the generalization error of SVMs, although this may be compensated partially by using more samples during training. SVMs are really useful due to their speed and general ease of use.

3) Naive Bayes Classifier

The Naive Bayes Classifier is a classification technique based on Bayes' Theorem in Probability theory. A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The Bayes' theorem is defined as follows,

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{1}$$

Here, each of the terms may be explained as

$P(A \mid B)$	Posterior Probability: Probability of A happening given B has occurred
$P(B \mid A)$	Prior Probability: Probability of B happening given A has occurred
$P(A)$,$P(B)$	Individual Probabilities of A and B

For example, In order to predict whether a specific person would choose a specific dish given that the dish could be sweet or savoury; the first thing to do would be to calculate the frequencies of whether the person has had chosen given that the dish was sweet or savoury and also that of whether they not. Secondly given the probability of each possibility i.e. savoury or sweet given the 'chosen' 'not chosen' scenarios, the likelihood table is created so that one can work on the provided query by comparing probabilities and choosing the most probable one as the query output.

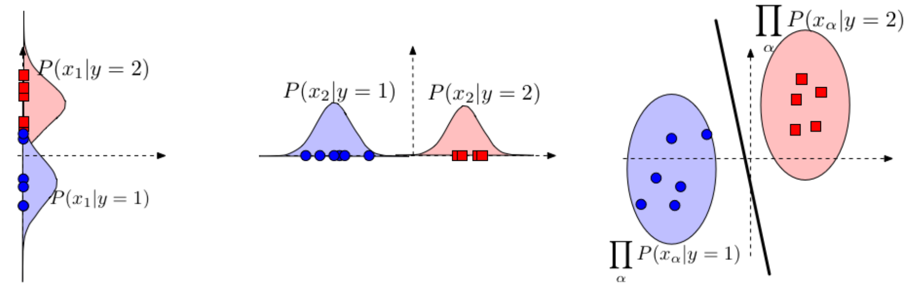

$P(x_1|y=2)$
$P(x_1|y=1)$
$P(x_2|y=1)$
$P(x_2|y=2)$
$\prod_\alpha P(x_\alpha|y=2)$
$\prod_\alpha P(x_\alpha|y=1)$

Figure 6: Naive Bayes Classifier infographic

3.5 Hardware and Software Requirements

3.5.1 Software Requirements

Software requirements may be summarized as below:

1. OS: Windows 7 or above / OSX / Linux / FreeBSD
2. Nginx >= 1.16.1
3. Python >= 3.8.0. Subdependencies:
   (a) numpy>=1.19.5
   (b) matplotlib>=3.4.2
   (c) numpy>=1.19.5
   (d) opencv-contrib-python>=4.5.2.54
   (e) pandas>=1.2.4
   (f) PyQt5>=5.15.4
   (g) scikit-learn>=0.24.2
   (h) seaborn>=0.11.1

(i) pandas>=1.3.0

(j) Flask>=2.0.1

These software requirements ease up if using Docker, and reduces to the list below. Here, any O/S supporting Docker or any container infrastructure may be used, in order to abstract the dependencies away.

1. Docker >= v20
2. Docker Compose >= v3.4

3.5.2 Hardware Requirements

The hardware requirements for this solution is rather simple, requiring moderately powerful hardware, which can be summarized as:

1. Processor: Intel Core i5 6500 and above (or AMD equivalent)
2. RAM: 16GB DDR3 (and equivalent) and above
3. Disk: 10GB+ Disk space (Either solid state or physical drive based)

4 Implementation

4.1 Languages/Tools/APIs used

The language used for implementing the various algorithms was Python. It is a popular multi-paradigm language that offers concise and readable code. Although Machine Learning is often filled with complex algorithms and diverse workflows, Python has a multitude of libraries that offer (relatively) high performance solutions to Machine Learning, letting the developer build reliable systems. There are many well-developed machine learning libraries and frameworks available with python that allow for complex systems to be build quickly. Out of them, `sklearn` and `cv2` are very well known solutions for Machine Learning and Image-Processing techniques. Interestingly, `cv2` is not a pure python library, but is a FFI to the C/C++ OpenCV2 libraries, which lend themselves well to creating rather responsive solutions. Additionally, `matplotlib` and `seaborn` was used to provide graphs and heatmaps for better visualization of statistics, and finally, `flask` was used as a microservice platform.

For the UI, a web application is most favourably built with Javascript, and here, the Next.JS (`next`) framework was used. The framework is a popular React-based framework to create statically exportable sites that can be served quick. Additionally, for UI Components, Bootstrap was used to provide pre-styled UI components and a well-made grid framework that lends itself to modern responsive design.

For platform management, an (optional) tool is used, Docker, and Docker Compose. Docker is a popular OS-level virtualization and containerization framework that can help orchestrate and manage complex bundled software. Docker Compose, is an addon, that allows for managing multiple containers at once, and is often useful in maintaining multi-container infrastructure, and can provide functionality for scaling and load balancing.

4.2 Use cases

Our project finds uses in examination and comparison of object detection and image recognition techniques in Machine Learning. The dataset supplied had roughly ˜500 images per class and all objects were sign languages. Another class was present for No type, although there were fewer images per class. The comparison here is applicable to areas of multi-class classification. Additionally, the solution should prove to be useful for quick classification of image data that contain Indian Sign Language.

4.3 Workflow Diagrams

The process for working on 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.

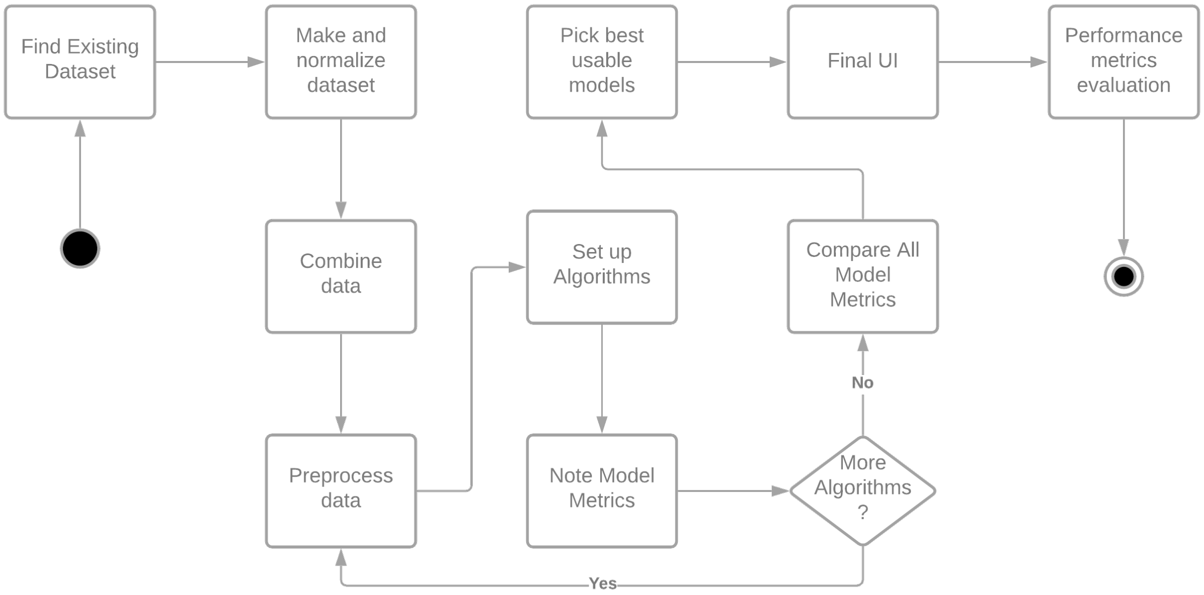


Figure 7: Workflow

In this procedure, three prominent classification algorithms would be set up and trained viz. k-Nearest Neighbour, Support Vector Machines and Naive Bayes classifier. After noting down the metrics for these models a comparison follows through resulting in the election of the algorithm with the most efficiency. After the model is set up, there's a need to develop a UI for ease of use. Lastly a final performance evaluation is carried out.

4.4 Data Preprocessing

The dataset used for training the models was a combination of a few datasets for Indian Sign Languages, one of them being custom-created. The annotation for the custom dataset was performed manually, and the data set was normalized to a 128x128 size. The whole dataset is then passed through a preprocessing pipeline, which is given as below

1. Skin Segmentation: Skin segmentation is used to retain the skin segment of a picture. Here, HSL thresholding was used to detect and conserve skin, and along with that, a single pass of morphological filtering was used to smooth out the image (Other methods seemed to have used multiple methods, but this was found to yield results that cropped out too much of the skin.)

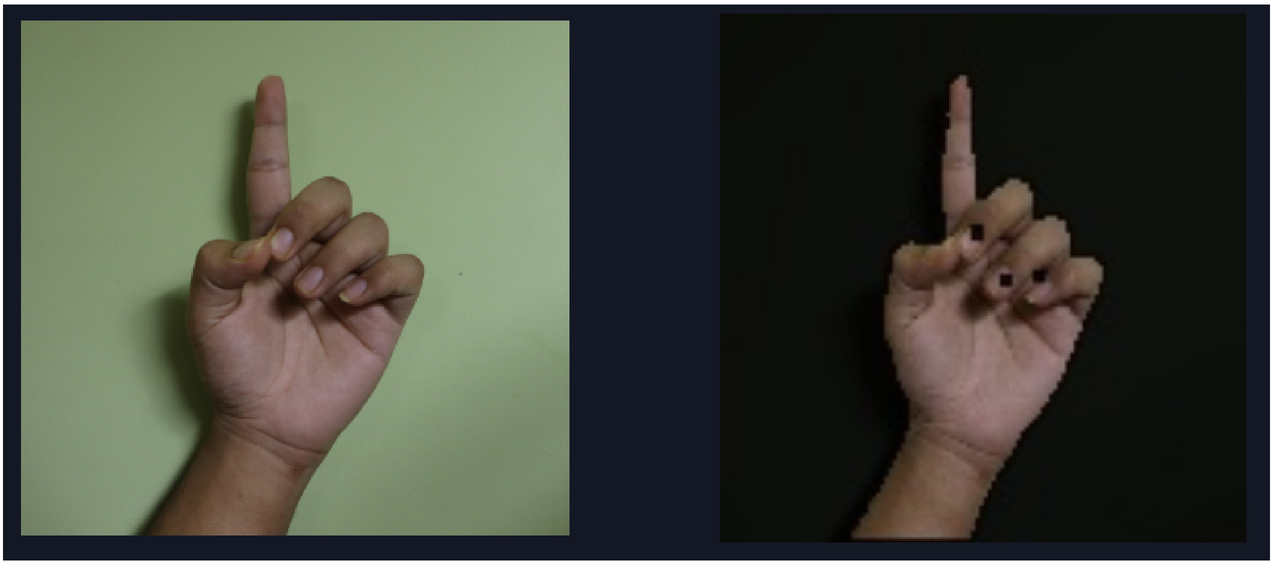


Figure 8: Resizing and skin segmentation

2. Feature detection: Feature detection is used to find the featureset in this image. Since the now skin-segmented image consists of only skin-based data, it should work better to identify features. Some algorithms have used SURF and ORB, however we picked up as a reliable and fast feature detection algorithm[4].

3. Clustering: The Visual Bag of Words concept is used here, to obtain a histogram, formed from the key points and descriptors as obtained in the previous steps. This yields key features and the resultant data which is actually used during classification.

4.5 Validation Methodology

During training, the models were provided with validation sets, which was roughly one-fourth the size of the training data set (Kept large due to comparatively low size). The testing of the model was done automatically by using the trained models to infer/predict the hand sign. The prediction results were also verified manually to ensure that the expected predictions were being made w.r.t the images. Additionally, to measure response timing, the inference timings for the combination of the 3 models were time for multiple requests to understand the delay that may be expected.

5 Result and Analysis

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

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 posed better accuracies, and theoretically, odd KNNs are easier to work with and present better accuracies as can resolve ties easier. For making the model, 7KNN was chosen, as higher K values tend to generalize further.

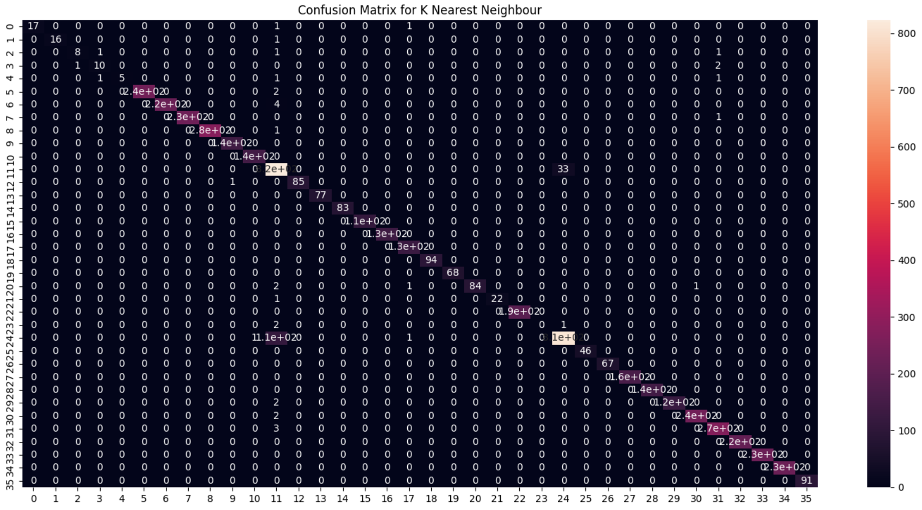


Figure 9: Confusion matrix for KNN

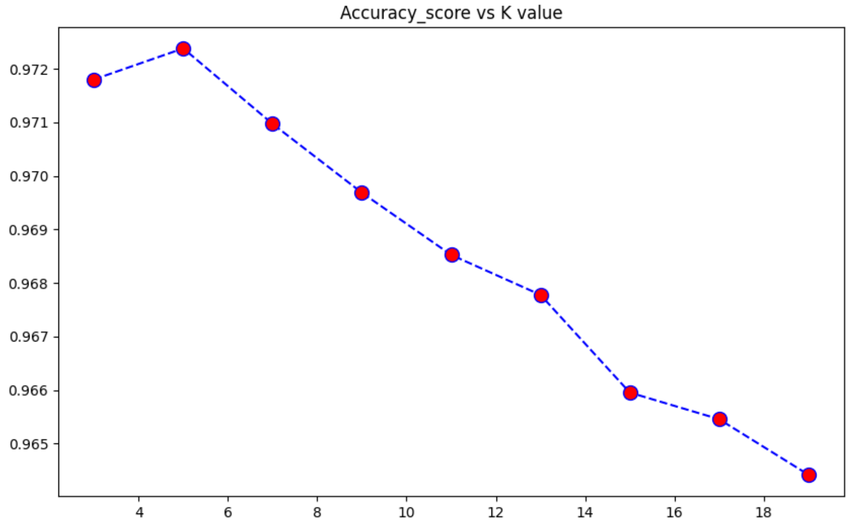
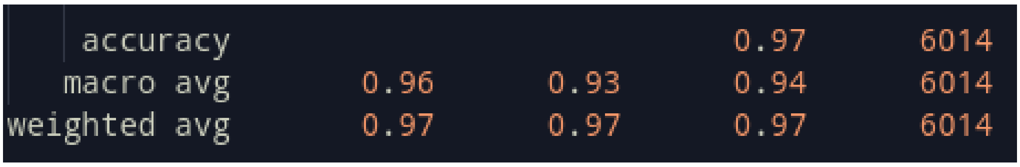


Figure 10: k vs Accuracy



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

Figure 11: 7KNN Statistics

The Figure 11 indicates the results for k=7 , as observed accuracy of 0.97.

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 increasing.

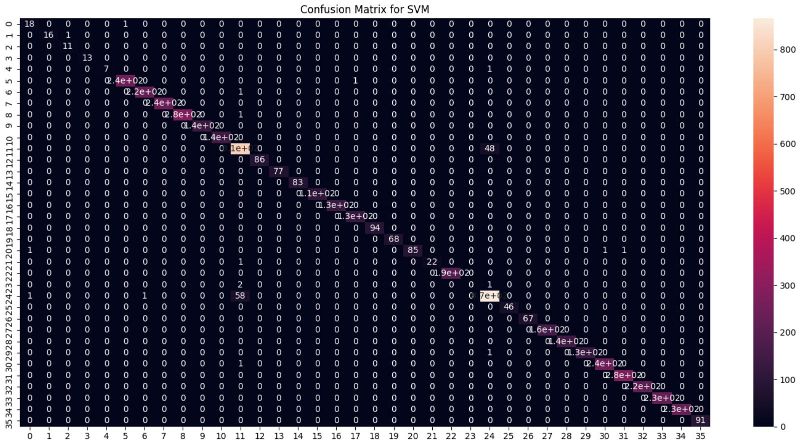
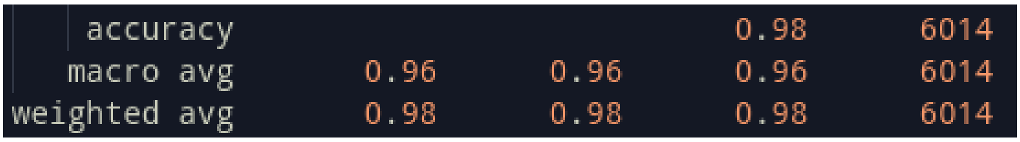


Figure 12: SVM confusion matrix



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

Figure 13: SVM Statistics

5.3 NBC

Three types of Naive Bayes' algorithm: The Gaussian form, Multinomial and Bernoulli's forms were considered. The Bernoulli's form performed the most reliably (while manual testing) and was chosen.

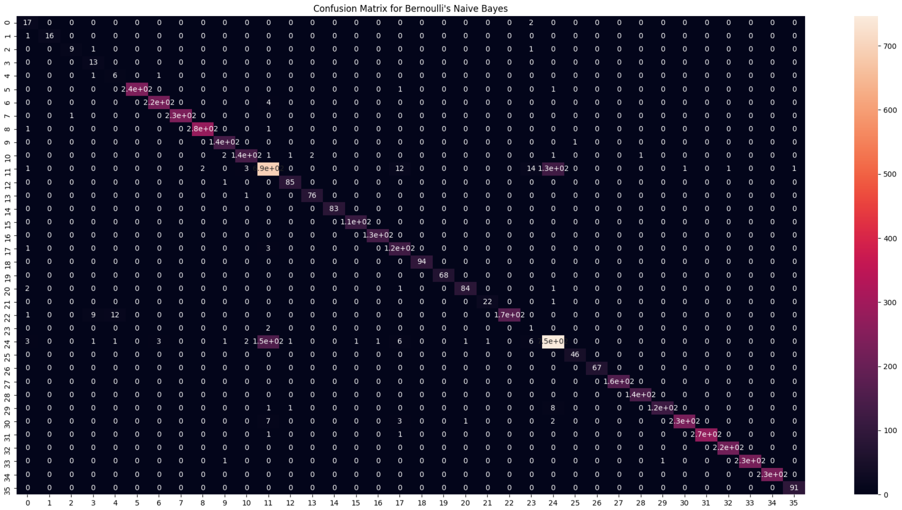
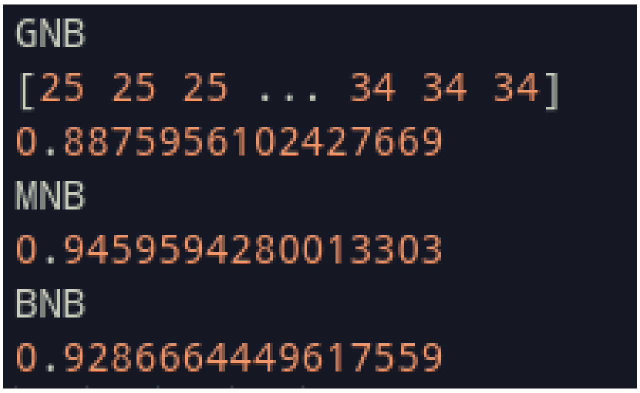
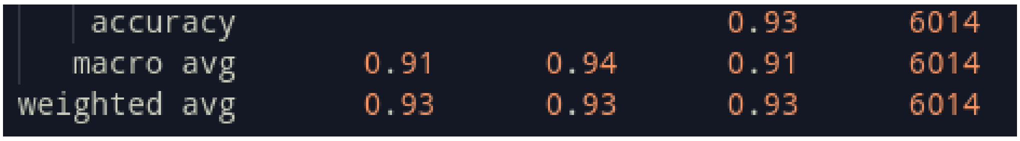


Figure 14: Bernoulli NBC confusion matrix



```
GNB
[25 25 25 ... 34 34 34]
0.8875956102427669
MNB
0.9459594280013303
BNB
0.9286644496175559
```

Figure 15: Accuracies of various NBC forms

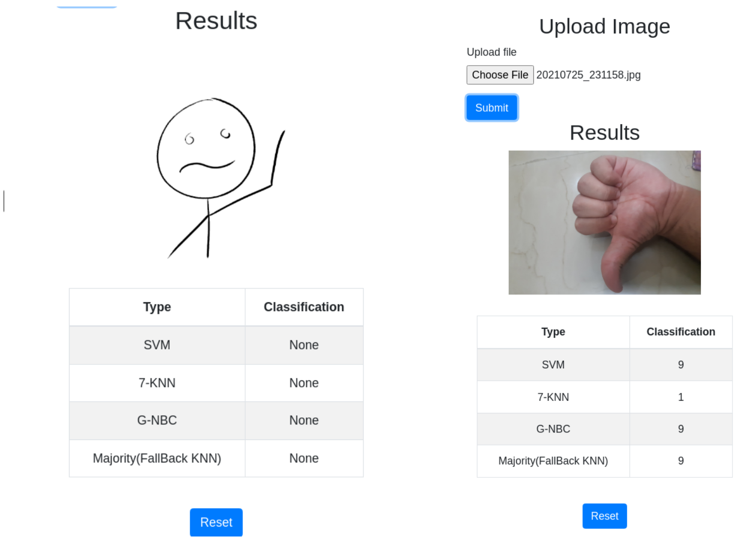


```
    accuracy                                0.93      6014
   macro avg         0.91        0.94       0.91      6014
weighted avg         0.93        0.93       0.93      6014
```

Figure 16: NBC Statistics

5.4 Overall Results

The overall results are summed up as follows, indicating two results, one for an irrelevant image and the other for a ’9’. As observed, all three algorithms show the output ’None’ for the first image as for the second, SVM and NBC were able to predict accurately, whereas kNN failed to classify due to its resemblance to the sign for 1.



Results

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

Reset

(a) None

Upload Image

Upload file

Choose File 20210725_231158.jpg

Submit

Results

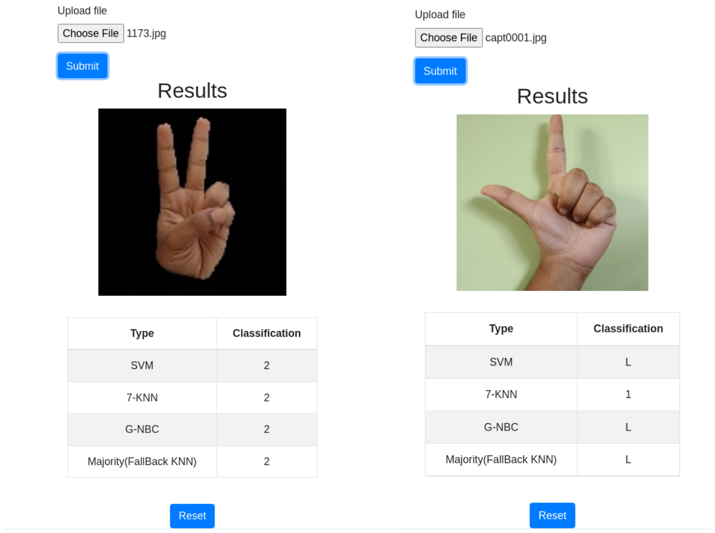
Type	Classification
SVM	9
7-KNN	1
G-NBC	9
Majority(FallBack KNN)	9

Reset

(b) 9

Figure 17: None and 9

Observing the images from below notice all three algorithms were successful to predict the sign for 2 although kNN again failed to indicate L because of resemblance to 1.



Upload file

Choose File 1173.jpg

Submit

Results

Type	Classification
SVM	2
7-KNN	2
G-NBC	2
Majority(FallBack KNN)	2

Reset

(a) 2

Upload file

Choose File capt0001.jpg

Submit

Results

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

Reset

(b) L

Figure 18: 2 and L

6 Conclusion and Future Enhancement

6.1 Novelty in the proposed solution

The motivation behind the project was a comparative study on the various models, and gain a deeper understanding of what would work better, and under what circumstances. Even considering this, novelty for the solution expands over the data set formation and usability:

- Inclusion of personal data inputs and establishing a new formed database for ISL.
- A hybrid data set which is an extension of existing sets with the current project inputs.
- Establishing a fresher and newer UI for public use.
- Producing a solution that can be scaled as per user requirements.

6.2 Limitations

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.

6.3 Future Enhancements

Even including the results, various future enhancements have been identified in the solution presented, that may help extending usability and performance of the solutions.

1. Adding hand detection: Tracking hands and identifying them would be a great boost to usability.
2. Using more varied datasets: Currently dataset sources are limited, and there is a lot of improvement that could be made with this regards.
3. Real time captioning : The performance of the models have shown that they can be reliably used in captioning. Given a way to train in other word-based gestures, real time captioning may be possible.

6.4 Conclusion

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 good solution for recognition of Indian Sign Language.

References

[1] 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.

[2] 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.

[3] Fernández Alcantarilla, Pablo & Bartoli, Adrien & Davison, Andrew. (2012). KAZE Features. 10.1007/978-3-642-33783-3_16.

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

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

[6] SirshenduHore, 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, Advancesin Intelligent Systems and Computing 455,DOI 10.1007/978-3-319-38771-0-54.

[7] M.A Hossen, Arun Govindaiah, Sadia Sultana and Alauddin Bhuiyan , "Bengali Sign Language Recognition Using Deep Convolutional Neural Network" ,2018 Joint7th International Conference on Informatics, Electronics - Vision (ICIEV) and 2018 2ndInternational Conference on Imaging, Vision & Pattern Recognition (icIVPR).

[8] Pradeep Kumar, Partha Pratim Roy and Debi Prosad Dogra, "Independent Bayesian classifier combination based sign language recognition using facial expression" , Elsevier Information SciencesArticle history:Received 7 September 2016,Revised 3 October2017,Accepted 23 October 2017,Available online 27 October 2017.

[9] Yue-cong Min, Aiming Hao, Xiujuan Chai and Xilin Chen , "Visual Alignment Constraint for Continuous Sign Language Recognition" , arXiv:2104.02330v1 [cs.CV]6 Apr 2021]

[10] AlinaKuznetsova, Laura Leal-Taixe and Bodo Rosenhah , "Real-time sign language recognition using a consumer depth camera" , Institute fuer Informationsverar-beitung, Leibniz University Hannover Appelstr. 9A, Hannover, 30167, Germany.

[11] Saipreethy.M.S,Valliammai. V and Padmavathi . S , "Indian Sign Language character recognition using neural networks", department of IT , Amrita vishwa vidyalaya.

[12] Huang,Wengang Zhou, Qilin Zhang, Houqiang Li and Weiping Li , "Video-Based Sign Language Recognition without Temporal Segmentation" , he Thirty-SecondAAAI Conference on Artificial Intelligence (AAAI-18).

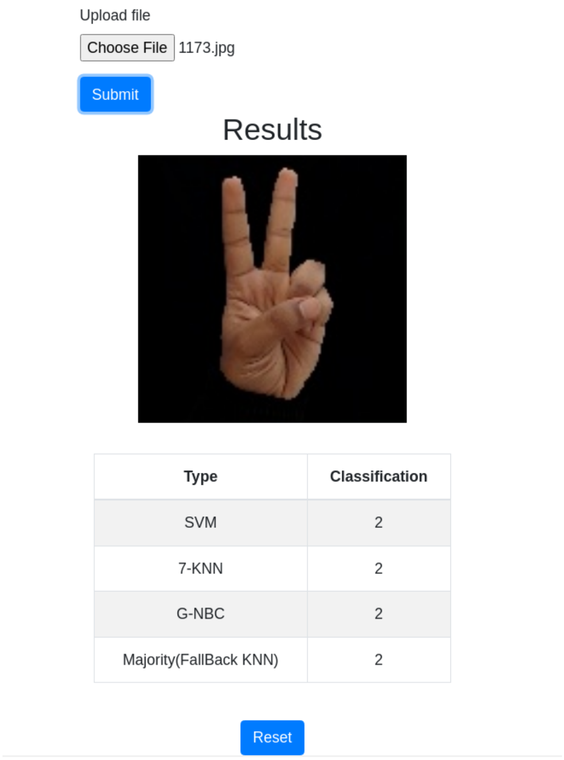
[13] Scar Kollerq, Simon Hadfield and Richard Bowden. "Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation" , IEEE XPLORE.

[14] Kirsti Grobel andMarcel1 Assan , "Isolated Sign Language Recognition using Hidden MarkovModels" IEEE Cybernetics conference.

[15] THAD STARNER AND ALEX PENTLAND ," Real Time ASL recognition from video using Hidden MarkovModel", MIT Boston.

[16] Eun-Jung Holden , Gareth Lee and Robyn Owens, "Australian sign language recognition" , Machine Vision and Applications (2005) 16(5): 312–320.

Appendix A



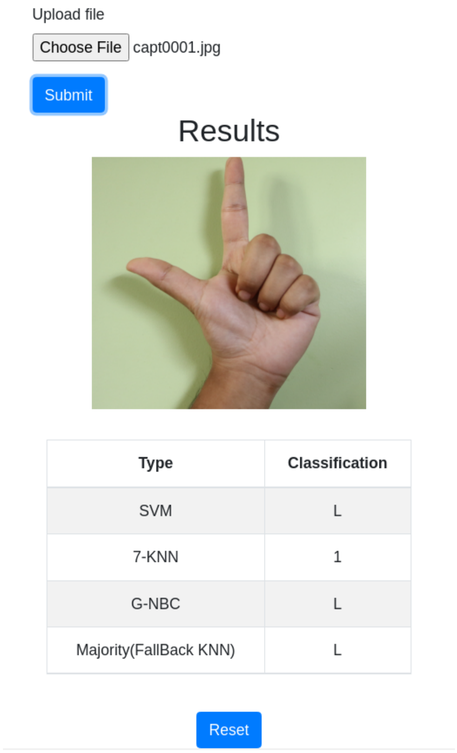
Upload file

1173.jpg

Results

Type	Classification
SVM	2
7-KNN	2
G-NBC	2
Majority(FallBack KNN)	2

Figure 19: 2



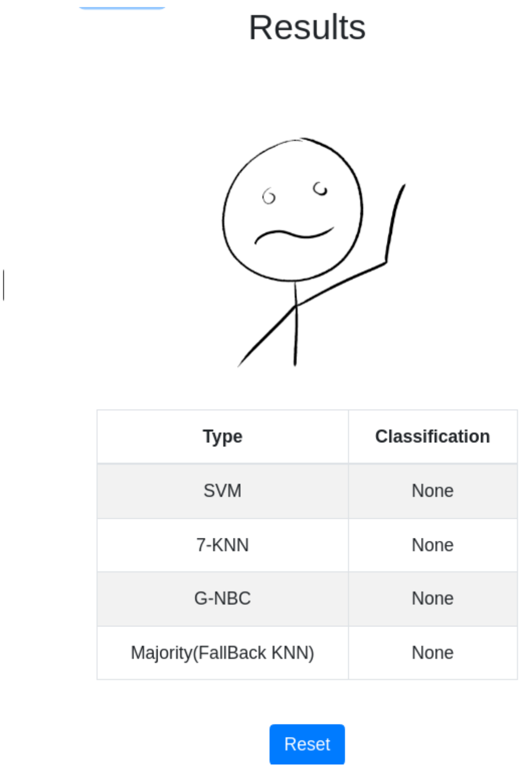
Upload file

capt0001.jpg

Results

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

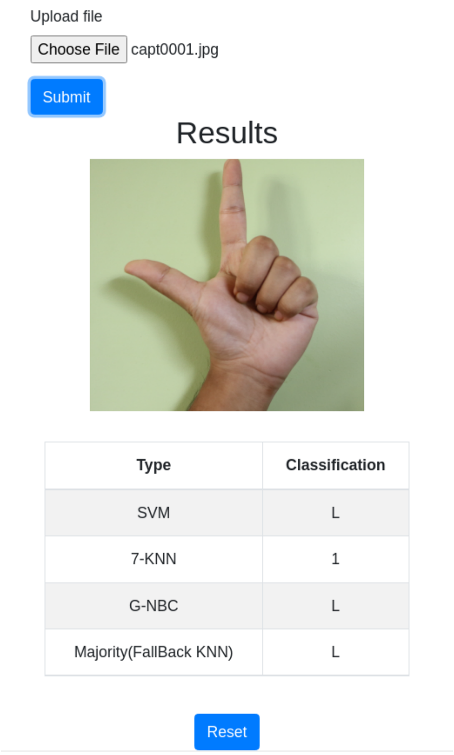
Figure 20: L



Results

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

Figure 21: None



capt0001.jpg

Results

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

Figure 22: L

Listing 1: KNN Output

```
For i= 3
For i= 5
For i= 7
For i= 9
For i= 11
For i= 13
For i= 15
For i= 17
For i= 19
Running for 7NN
0.9694047223145993
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        19
           1       1.00      0.94      0.97        17
           2       0.89      0.73      0.80        11
           3       0.83      0.77      0.80        13
           4       1.00      0.62      0.77         8
           5       1.00      0.99      1.00       240
           6       1.00      0.98      0.99       226
           7       1.00      1.00      1.00       235
           8       1.00      1.00      1.00       279
           9       0.99      1.00      1.00       136
          10       0.99      1.00      1.00       144
          11       0.86      0.96      0.91       856
          12       1.00      0.99      0.99        86
          13       1.00      1.00      1.00        77
          14       1.00      1.00      1.00        83
          15       1.00      1.00      1.00       111
          16       1.00      1.00      1.00       134
          17       0.98      1.00      0.99       127
          18       1.00      1.00      1.00        94
          19       1.00      1.00      1.00        68
          20       1.00      0.95      0.98        88
          21       1.00      0.96      0.98        23
          22       1.00      1.00      1.00       190
          23       0.00      0.00      0.00         3
          24       0.96      0.88      0.92       926
          25       1.00      1.00      1.00        46
          26       1.00      1.00      1.00        67
          27       1.00      1.00      1.00       158
          28       1.00      1.00      1.00       144
          29       1.00      0.98      0.99       127
          30       1.00      0.99      0.99       241
          31       0.98      0.99      0.99       275
          32       1.00      1.00      1.00       215
          33       1.00      1.00      1.00       228
          34       1.00      1.00      1.00       228
          35       1.00      1.00      1.00        91

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

Listing 2: SVM Output

```
0.9795477219820419
              precision    recall  f1-score   support

           0       0.90      0.95      0.92        19
           1       1.00      0.94      0.97        17
           2       0.92      1.00      0.96        11
           3       1.00      1.00      1.00        13
           4       1.00      0.88      0.93         8
           5       1.00      1.00      1.00       240
           6       1.00      1.00      1.00       226
           7       1.00      1.00      1.00       235
           8       1.00      1.00      1.00       279
           9       1.00      1.00      1.00       136
          10       1.00      1.00      1.00       144
          11       0.93      0.94      0.94       856
          12       1.00      1.00      1.00        86
          13       1.00      1.00      1.00        77
          14       1.00      1.00      1.00        83
          15       1.00      1.00      1.00       111
          16       1.00      1.00      1.00       134
          17       0.99      1.00      1.00       127
          18       1.00      1.00      1.00        94
          19       1.00      1.00      1.00        68
          20       1.00      0.97      0.98        88
          21       1.00      0.96      0.98        23
          22       1.00      1.00      1.00       190
          23       0.00      0.00      0.00         3
          24       0.94      0.94      0.94       926
          25       1.00      1.00      1.00        46
          26       1.00      1.00      1.00        67
          27       1.00      1.00      1.00       158
          28       1.00      1.00      1.00       144
          29       1.00      0.99      1.00       127
          30       1.00      1.00      1.00       241
          31       1.00      1.00      1.00       275
          32       1.00      1.00      1.00       215
          33       1.00      1.00      1.00       228
          34       1.00      1.00      1.00       228
          35       1.00      1.00      1.00        91

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

Listing 3: NBC Output

```
GNB
[25 25 25 ... 34 34 34]
0.8875956102427669
MNB
0.9459594280013303
BNB
0.9286664449617559
              precision    recall  f1-score   support

           0       0.63      0.89      0.74        19
           1       1.00      0.94      0.97        17
           2       0.90      0.82      0.86        11
           3       0.52      1.00      0.68        13
           4       0.32      0.75      0.44         8
           5       1.00      0.99      1.00       240
           6       0.98      0.98      0.98       226
           7       1.00      1.00      1.00       235
           8       0.99      0.99      0.99       279
           9       0.96      0.99      0.98       136
          10       0.96      0.94      0.95       144
          11       0.80      0.81      0.81       856
          12       0.97      0.99      0.98        86
          13       0.97      0.99      0.98        77
          14       1.00      1.00      1.00        83
          15       0.99      1.00      1.00       111
          16       0.99      1.00      1.00       134
          17       0.84      0.97      0.90       127
          18       1.00      1.00      1.00        94
          19       1.00      1.00      1.00        68
          20       0.98      0.95      0.97        88
          21       0.96      0.96      0.96        23
          22       1.00      0.88      0.94       190
          23       0.04      0.33      0.07         3
          24       0.84      0.81      0.82       926
          25       0.98      1.00      0.99        46
          26       1.00      1.00      1.00        67
          27       1.00      1.00      1.00       158
          28       0.99      1.00      1.00       144
          29       0.99      0.92      0.96       127
          30       1.00      0.95      0.97       241
          31      1.00      0.99      1.00       275
          32      1.00      1.00      1.00       215
          33      1.00      0.99      1.00       228
          34      1.00      1.00      1.00       228
          35      0.99      1.00      0.99        91

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

Appendix B