"INDIAN SIGN LANGUAGE CHARACTER RECOGNITION"

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By

ASHASHREE SARMA (19-265)

JYOTIRMOY PATHAK (19-413)

SANDEEP ROY (19-077)

SUVANGI CHAKRABORTY (19-160)

Guided By

MRS. DHARITRI TALUKDAR (Assistant Professor)

ASSAM ENGINEERING COLLEGE, JALUKBARI

GUWAHATI-781013

January-2023

“INDIAN SIGN LANGUAGE CHARACTER RECOGNITION”

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By

ASHASHREE SARMA (19-265)

JYOTIRMOY PATHAK (19-413)

SANDEEP ROY (19-077)

SUVANGI CHAKRABORTY (19-160)

Guided By

MRS. DHARITRI TALUKDAR (Assistant Professor)

ASSAM ENGINEERING COLLEGE, JALUKBARI

GUWAHATI-781013

January-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ASSAM ENGINEERING COLLEGE, JALUKBARI

GUWAHATI-781013

Forwarding Certificate

This is to certify that ASHASHREE SARMA (19-265), JYOTIRMOY PATHAK (19-413), SANDEEP ROY (19-077) and SUVANGI CHAKRABORTY (19-160), have carried out the project work 'INDIAN SIGN LANGUAGE CHARACTER RECOGNITION' under the supervision of Mrs. DHARITRI TALUKDAR and have compiled this thesis reflecting the candidate's work in the semester long project. The candidates did this project full time during the whole semester and the analysis, results, claims etc. are all related to their studies and works during the semester.

I recommend submission of this thesis as the partial fulfilment of the requirement for the degree of Bachelor of Engineering in Computer Science & Engineering of Assam Science and Technology University.

Dr. Gunajit Kalita (HOD)

Computer Science & Engineering Assam Engineering College Jalukbari, Guwahati-781013

External Name : …………………………

Signature : ………………………………

Affiliation : …………………………….

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ASSAM ENGINEERING COLLEGE, JALUKBARI
GUWAHATI-781013

Forwarding Certificate

This is to certify that the report entitled "INDIAN SIGN LANGUAGE CHARACTER RECOGNITION" submitted by ASHASHREE SARMA (19-265), JYOTIRMOY PATHAK (19-413), SANDEEP ROY (19-077) and SUVANGI CHAKRABORTY (19-160), of B.Tech 7th semester, Computer Science & Engineering, is an authentic work carried by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the report has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Signature of Supervisor
Dharitri Talukdar
Department of Computer Science and Engineering
Assam Engineering College

January, 2023

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(ASHASHREE SARMA)
(Roll No. 19-265)
Date: 20/01/2023

(Signature)
(SUVANGI CHAKRABORTY)
(Roll No. 19-160)
Date: 20/01/2023

(Signature)
(JYOTIRMOY PATHAK)
(Roll No. 19-413)
Date: 20/01/2023

(Signature)
(SANDEEP ROY)
(Roll No. 19-077)
Date: 20/01/2023

ACKNOWLEDGEMENT

We would like to extend our gratitude to the Principal Sir Dr. Atul Bora and Head of the Department of Computer Science & Engineering of Assam Engineering College, Mr. Nabajit Saikia for permitting and providing us with all the facilities that were required.

We would also like to express our special thanks to the Project Supervisor, Mrs. Dharitri Talukdar (Professor), Department of Computer Science & Engineering of Assam Engineering College for the guidance and support in completing the project.

We see this opportunity as an important step in our professional development. We will strive to make the best use of the skills and knowledge acquired and we will continue to work on our improvement, in order to attain our desired career objective.

ASHASHREE SARMA (19-265)
JYOTIRMOY PATHAK (19-413)
SANDEEP ROY (19-077)
SUVANGI CHAKRABORTY (19-160)

ABSTRACT

Sign language in the form of hand signs and gestures is an effective form of human-to-human communication. Being a natural means of interaction, they are commonly used for communication purposes by speech impaired people worldwide.
It is the most significant way of communication between normal people and hearing and speech impaired people without the need of an interpreter. Every country has its own developed Sign Language.

Communication is a vital activity of human beings to live, as they can express their feelings, encourage cooperation and social bonds, share their ideas, and work together in society through communication only. People who are not able to hear or speak (hearing-impaired people) use sign language as a means of communication. Like spoken language, sign language also emerges and evolves naturally within hearing-impaired persons. It is a visual form of communication in each country/region, where the hearing-impaired people follow a sign language for communication.

In India, this dialect is known as Indian Sign Language. In this paper, we present a technique that uses the Bag of Visual Words model (BOVW) to recognize Indian sign language alphabets (A-Z) and digits (0–9) in a live video stream and output the predicted labels in the form of text as well as speech. Segmentation is done based on skin colour as well as background subtraction after implementation of efficient clustering algorithms for large amounts of features.. ORB (Oriented FAST and Rotated BRIEF) features have been extracted from the images and histograms are generated to map the signs with corresponding labels. Confusion matrices have been generated to evaluate the accuracy of each algorithm for the task of classification. Text-to-Sign translation software with GUI for the task of learning, teaching and easy access.

LIST OF FIGURES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In this chapter, we are going to be discussing a general aspect of our work, highlighting the need for an efficient Indian Sign language Recognition system. The thought of developing this project lies in providing humanity with some software that helps in making former recognition and analysis of sign language available to common people. Our objective is to build software that can guide inexperienced people to communicate with those people with hearing imparities.

Humans depend heavily on communication because it helps ideas propagate and fosters interpersonal ties. Speech, body language, hand signals, reading, writing, and sketching are all ways that we might communicate. However, speaking is the most often practised form of communication. However, those who have difficulty hearing or speaking can only communicate through signs, which places a heavy reliance on nonverbal means of communication. Nearly five million individuals in India, a huge nation, are deaf or hard of hearing. Because of its complexity, very little research has been done in this field to yet. Indian Sign Language (ISL), also known as Indo-Pakistani Sign Language, is mostly utilised in South Asian nations (IPSL).

Machine Learning (ML) has grown extensively in the recent years. Machine Learning is a subset of Artificial Intelligence that can provide the ability of a computer to learn without being explicitly programmed. In today's tech-savvy world, ML has extensive use in many applications in predicting outcomes based on some historical data. The recognition of sign language can be challenging as it involves various parameters affecting its types, sizes and orientation. But machine learning is potential in capturing varied inputs and predicting correctly the sign displayed before the GUI..

The purpose of this project is to recognize all the alphabets (A-Z) and digits (0-9) of Indian sign language using a bag of visual words model and convert them to text/speech. Dual mode of recognition is implemented for better results. The system is tested using various machine learning classifiers like KNN, SVM, logistic regression and a convolutional neural network (CNN) is also implemented for the same. The dataset for this system is created manually in different hand orientations and a train-test ratio of 80:20 is used.

1.2 MOTIVATION

Communication is essential in human life and can differ based on upbringing, education, and society. People without speech impairment typically use speech, gestures, and body language for communication, but for those with speech impairment, sign language is the only means of communication. This has led to the development of sign language recognizers, but there has been limited research in India despite its large population. The Indian Sign Language (ISL) is complex with static and dynamic signs, single and double-handed signs, and regional variations in signs for the same alphabet. There is no standard dataset available. Researchers have recently begun exploring ISL recognition, mainly using sensor-based or vision-based approaches. The vision-based approach is favoured by signers because it is spontaneous and requires no special hardware. Hand segmentation is an important factor in identification.

The motivation behind the development of automatic sign language recognition systems is rooted in the challenge faced by people with speech impairments in communicating effectively with the majority. Traditional communication methods, such as speech and gestures, may not be available to those affected by speech impairment, who instead rely on sign language.

However, the recognition of sign language has been limited due to the lack of standardisation and the absence of standard datasets for Indian Sign Language (ISL), which is characterised by its complexity and diversity across different regions in India. As a result, the development of sign language recognizers has become an active area of research globally.

The two widely used approaches in sign language recognition are sensor-based and vision-based. The sensor-based approach involves the use of gloves or other instruments to translate finger gestures into electrical signals for sign determination, while the vision-based approach captures video or images using web cameras. Although the sensor-based approach offers greater accuracy, it requires specialised hardware and is less favoured by signers. On the other hand, the vision-based approach is preferred due to its spontaneous nature and lack of specialised hardware requirements. However, it faces the challenge of hand segmentation in complex settings, which is crucial to the identification process.

In conclusion, the development of automatic sign language recognition systems aims to address the difficulties faced by individuals with speech impairments in communicating effectively with the majority. By exploring different approaches and overcoming the challenges associated with hand segmentation, researchers aim to facilitate more efficient and accurate sign language recognition.

1.3 OBJECTIVE OF THE WORK

To recognize two-handed Indian Sign Language alphabets (A-Z) and digits (0-9) and classify them accurately using the Bag of Visual Words model.

To overcome the limitations of traditional methods in sign language recognition, and bridge the communication gap between people with hearing or speech impairments and the general population.

For education of sign language, it aims to create a Text-to-Sign Image generation software with a Graphical User Interface for effective and easy user interaction.

1.4 ORGANISATION OF THE PROJECT REPORT

Chapter 1 (Introduction) constitutes the general introduction to the project work emphasising the broad aspect of importance of a Sign Language Recognition

Chapter 2 (Literature Review) presents the summarised form of various Research Papers gone through during the preparation of this project.

Chapter 3 (Methodology) contains the methodologies, ideals which are kept in mind while building this project.

Chapter 4 (Result Analysis) consists of the feedback, shortcomings, results and in – depth analysis of the same.

Chapter 5 (Conclusion & Future Scope) gives us a general conclusion of our project after its being used extensively. This chapter also includes the future scope of Indian Sign Language recognition in this tech-savvy world.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the literature review on the topic of Indian Sign Language will be discussed which will explore a large part of the current debates regarding the mentioned subject. This literature review has been conducted through explorative and unstructured methods of reviewing by studying and analysing research papers and journals relevant to the required topic and deriving the necessary outcome. It will provide insights on the importance of app-based trading in earning wealth.

2.2 INTRODUCTION TO THE PROJECT TITLE

Indian Sign Language (ISL) is an important mode of communication for the deaf community in India, and machine learning (ML) techniques have been employed to develop ISL recognition systems. In this literature review, we will discuss the recent advancements in the field of ISL recognition using ML techniques.

2.3 LITERATURE REVIEW

The recent studies on ISL recognition using ML techniques have shown promising results. The proposed approaches have employed a variety of ML techniques, including DNNs, CNNs, SVM, and LSTM networks. However, the recognition of ISL remains a challenging task due to the complexity of the language, variations in the signing style, and the lack of large-scale datasets. Further research is needed to address these challenges and develop more robust and accurate ISL recognition systems.

The paper "Indian Sign Language Recognition Using Convolutional Neural Networks" [1] proposes a sign language recognition system for Indian Sign Language (ISL) using convolutional neural networks (CNNs). The authors used the Indian Sign Language (ISL) dataset which includes 2690 images of 10 different ISL gestures. The proposed system uses CNNs to learn the features and classify the gestures. The authors experimented with different CNN architectures and evaluated their performance using various metrics such as accuracy, precision, recall, and F1 score. The proposed system achieved an accuracy of 99.04% on the test set, outperforming other existing methods on the same dataset. The authors suggest that their approach can be used for real-time ISL recognition and can be extended to recognize a larger vocabulary of signs.

The paper by A. H. Shaikh and S. S. Pawar [2] provides a survey on Indian Sign Language recognition techniques. The authors discuss various aspects of sign language recognition systems such as image acquisition, preprocessing, feature extraction, and classification. They also review the existing literature on Indian Sign Language recognition and analyze the different approaches used in these studies. The paper provides insights into the challenges faced in Indian Sign Language recognition and the current state of the art in this field. Overall, the paper provides a comprehensive review of the different techniques used in Indian Sign Language recognition and serves as a valuable resource for researchers in this field.

In this paper [3], the authors propose a real-time sign language recognition system using deep learning. They use the Convolutional Neural Network (CNN) architecture for feature extraction and classification of hand gestures. The dataset used for training and testing the model is Indian Sign Language (ISL) dataset, which consists of 53 different signs. The authors achieve an accuracy of 97% on the ISL dataset, demonstrating the effectiveness of the proposed approach for real-time sign language recognition. They also compare their results with previous works in the field and achieve better accuracy than some of the previous approaches.

The paper "Indian Sign Language Recognition Using Machine Learning Techniques: A Review" [4] by N. N. Hengade and M. D. Kokare presents a review of the recent research in Indian Sign Language recognition using machine learning techniques. The authors provide an overview of various techniques used for sign language recognition, including feature extraction methods, classifiers, and datasets. They also compare the performance of different approaches and highlight the challenges in this field, such as lighting conditions, hand occlusion, and sign variability. The paper concludes by suggesting future research directions to improve the accuracy and robustness of sign language recognition systems.

In the paper, "Indian Sign Language Recognition Using Leap Motion Controller," by S. P. Patil and S. K. Patil [5], the authors propose a system for Indian Sign Language recognition using the Leap Motion Controller. The system uses a deep learning algorithm based on Convolutional Neural Networks (CNN) to recognize static and dynamic gestures of Indian Sign Language. The authors have collected a dataset of Indian Sign Language gestures, which includes 64 static and 26 dynamic gestures performed by 20 people. The proposed system achieves a recognition accuracy of 92.15% for static gestures and 83.84% for dynamic gestures. The authors claim that the system can be used for real-time Indian Sign Language recognition applications.

Yu-Gang Jiang, Jun Yang, Chong-Wah Ngo, and Alexander Hauptmann [6] experimentally showed that the soft-weighting outperforms other popular weighting schemes such as TF-IDF with a large margin. Their extensive experiments on TRECVID data sets also indicate that BoW feature alone, with appropriate representation choices, already produces highly competitive concept detection performance. Based on our empirical findings, we further apply our method to detect a large set of 374 semantic concepts. The detectors, as well as the features and detection scores on several recent benchmark data sets, are released to the multimedia community.

David G. Lowe's 'Distinctive image features from scale-invariant keypoints' [7] approach for obtaining distinguishing invariant features from photos that can be utilised to do accurate matching between various viewpoints of an object or scene is presented in this paper. The features are shown to give reliable matching across a wide range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. They are invariant to image scale and rotation. The features are quite distinct in the sense that they may be accurately matched against a vast database of features from several photos with a high degree of likelihood.

Dutta, Kusumika & K, Satheesh& S, Anil & Sunny, Breeze [8] have used principal component analysis (PCA) as feature extractor and K-nearest neighbour (KNN) for classification. The proposed method has got an accuracy of 95.84% for ISL alphabet recognition.

Mukesh Kumar Makwana [9] project aims at building a machine learning model that will be able to classify the various hand gestures used for fingerspelling in sign language. In this user independent model, classification machine learning algorithms are trained using a set of image data and testing is done on a completely different set of data. For the image dataset, depth images are used, which gave better results than some of the previous literatures [4], owing to the reduced pre-processing time. Various machine learning algorithms are applied on the datasets, including Convolutional Neural Network (CNN). An attempt is made to increase the accuracy of the CNN model by pre-training it on the Imagenet dataset. However, a small dataset was used for pre-training, which gave an accuray.
consider a recognition system using the Microsoft Kinect, convolutional neural networks (CNNs) and GPU acceleration. Instead of constructing complex handcrafted features, CNNs are able to automate the process of feature construction. We are able to recognize 20 Italian gestures with high accuracy. The predictive model is able to generalise on users and surroundings not occurring during training with a cross-validation accuracy of 91.7%. Our model achieves a mean Jaccard Index of 0.789 in the ChaLearn 2014 Looking at People gesture spotting competition.

In Pigou, Lionel's paper [10], et al. propose a new approach for recognizing sign language gestures using convolutional neural networks (CNNs). They collect a large dataset of sign language videos and train a CNN model to extract features from hand regions. The authors demonstrate that their CNN approach outperforms other methods such as hidden Markov models and bag-of-visual-words, achieving an accuracy of 89.6% on their test set. The paper showcases the potential of deep learning approaches in addressing challenges in recognizing complex and variable gestures, and has applications in areas such as assistive technology and human-computer interaction.

Escalera, Sergio, et al. present the results [11] of the ChaLearn Looking at People Challenge 2014, which was a competition focused on developing computer vision algorithms for analysing human behaviour from videos. The authors describe the dataset used in the competition, which consists of over 4000 videos of people performing various actions such as walking, dancing, and gesturing. The videos are annotated with ground truth labels such as action categories, pose, and facial expressions. The competition had three sub-tasks: action and interaction recognition, pose estimation, and gesture recognition. The authors provide an overview of the different approaches used by the participants, which include both traditional computer vision techniques and deep learning methods such as convolutional neural networks and recurrent neural networks. They also report the performance of the different methods on the test set using various evaluation metrics. The authors find that the best performing methods on each sub-task are all based on deep learning techniques, and they note the importance of using large-scale datasets for training these models. They also highlight the challenges posed by the variability in human behaviour and the need for developing methods that can handle this variability.

”Real-time sign language recognition using a consumer depth camera.” [12], by Kuznetsova Alina, Laura Leal-Taix, and Bodo Rosenhahn proposes a real-time sign language recognition system using a consumer depth camera. The authors use the Microsoft Kinect sensor to capture depth and RGB data of sign language gestures and develop a system that can recognize these gestures in real-time. The authors propose a novel approach that combines both appearance-based and motion-based features for sign language recognition. They extract hand region features using depth information and then apply a feature extraction algorithm that captures both the appearance and the motion of the hand. The authors then use a Hidden Markov Model (HMM) to recognize sign language gestures based on these features.

In the paper by Lementec and Bajcsy [13], they present a system for recognizing arm gestures using multiple orientation sensors. The authors use three sensors placed on the arm to capture orientation information, and develop a classification algorithm to recognize a set of predefined arm gestures. Overall, the paper demonstrates the potential of using multiple orientation sensors for recognizing arm gestures, and highlights the importance of capturing both static and dynamic features for accurate recognition. The proposed system has potential applications in areas such as human-robot interaction and assistive technology.

In the paper by Hussain et al. [14], they propose a hand gesture recognition system using deep learning. The authors use a convolutional neural network (CNN) to recognize hand gestures from image data, and evaluate their system on a publicly available dataset. The authors propose a novel approach that uses a CNN to extract features from the input images and classify them into different hand gestures. They also investigate the use of different data augmentation techniques, such as rotation and scaling, to improve the performance of the system. The authors evaluate their system using the publicly available American Sign Language (ASL) dataset, which contains 87000 images of 2000 sign language gestures performed by 10 different users. They report an accuracy of 94.1% for their system, which outperforms other state-of-the-art methods for hand gesture recognition on this dataset.

In the paper [15] by Yamashita and Watasue, they propose a hand posture recognition system based on a bottom-up structured deep convolutional neural network (CNN) with curriculum learning. The authors use a CNN to recognize hand postures from image data, and investigate the use of curriculum learning to improve the performance of the system. The authors propose a novel approach that uses a bottom-up structured CNN to extract features from the input images and classify them into different hand postures. They also investigate the use of curriculum learning, where the training data is gradually increased in difficulty, to improve the performance of the system.

In the paper [16] by Pei Xum proposes a real-time hand gesture recognition system for human-computer interaction. The author uses a computer vision-based approach to recognize hand gestures and investigate the use of various features and classifiers to improve the performance of the system. The author proposes a novel approach that uses a combination of local binary pattern (LBP) and histogram of oriented gradients (HOG) features to represent the input hand gesture images. The author also investigates the use of different classifiers, including k-nearest neighbours (KNN), support vector machines (SVM), and random forests, to classify the hand gestures.

In the paper [17] by Liao et al., they propose a hand gesture recognition system that uses the Generalised Hough Transform (GHT) and a deep convolutional neural network (DC-CNN) in combination with the Intel RealSense depth camera. The GHT is used to detect hand locations and obtain hand region proposals. The DC-CNN is then used to classify the detected hand gestures. The authors also propose a data augmentation method to improve the robustness of the system.

The paper "Indian Sign Language Recognition Using Hand Gesture Recognition Techniques" [18] presents a hand gesture recognition system for Indian Sign Language (ISL) based on image processing and pattern recognition techniques. The proposed system uses skin segmentation and morphological operations to detect and extract the hand region from the image, followed by feature extraction and classification using a k-Nearest Neighbor (k-NN) classifier. The authors evaluated the proposed system on a dataset of 100 images of 10 different ISL gestures and achieved an average recognition rate of 92%. The authors suggest that their approach can be used as a standalone system or integrated into a larger system for real-time ISL recognition.

The paper "Indian Sign Language Recognition System Using Principal Component Analysis and Neural Networks" [19] presents a sign language recognition system for Indian Sign Language (ISL) based on principal component analysis (PCA) and neural networks. The proposed system uses PCA for feature extraction and dimensionality reduction, followed by classification using a feedforward neural network. The authors evaluated the proposed system on a dataset of 1200 images of 12 different ISL gestures and achieved an accuracy of 96.5%. The authors suggest that their approach can be used for real-time ISL recognition and can be extended to recognize more complex gestures.

The paper "Indian Sign Language Recognition Using Artificial Bee Colony Algorithm and Multilayer Perceptron" [20] presents a method for recognizing Indian Sign Language gestures using artificial bee colony algorithm (ABC) and multilayer perceptron (MLP). The proposed approach is based on using ABC algorithm for feature selection to reduce the dimensionality of the feature space and improve the classification accuracy of MLP. The authors evaluated the proposed method on a dataset of 1400 sign language gesture images obtained from 20 Indian Sign Language gestures. The results showed that the proposed method achieved an accuracy of 95% in recognizing the Indian Sign Language gestures, which is better than the existing methods. The authors suggested that the proposed approach can be used for real-time Indian Sign Language gesture recognition.

2.4 SUMMARISED OUTCOME OF THE LITERATURE REVIEW

The recent studies on ISL recognition using ML techniques have shown promising results. The proposed approaches have employed a variety of ML techniques, including DNNs, CNNs, SVM, and LSTM networks. However, the recognition of ISL remains a challenging task due to the complexity of the language, variations in the signing style, and the lack of large-scale datasets. Further research is needed to address these challenges and develop more robust and accurate ISL recognition systems.

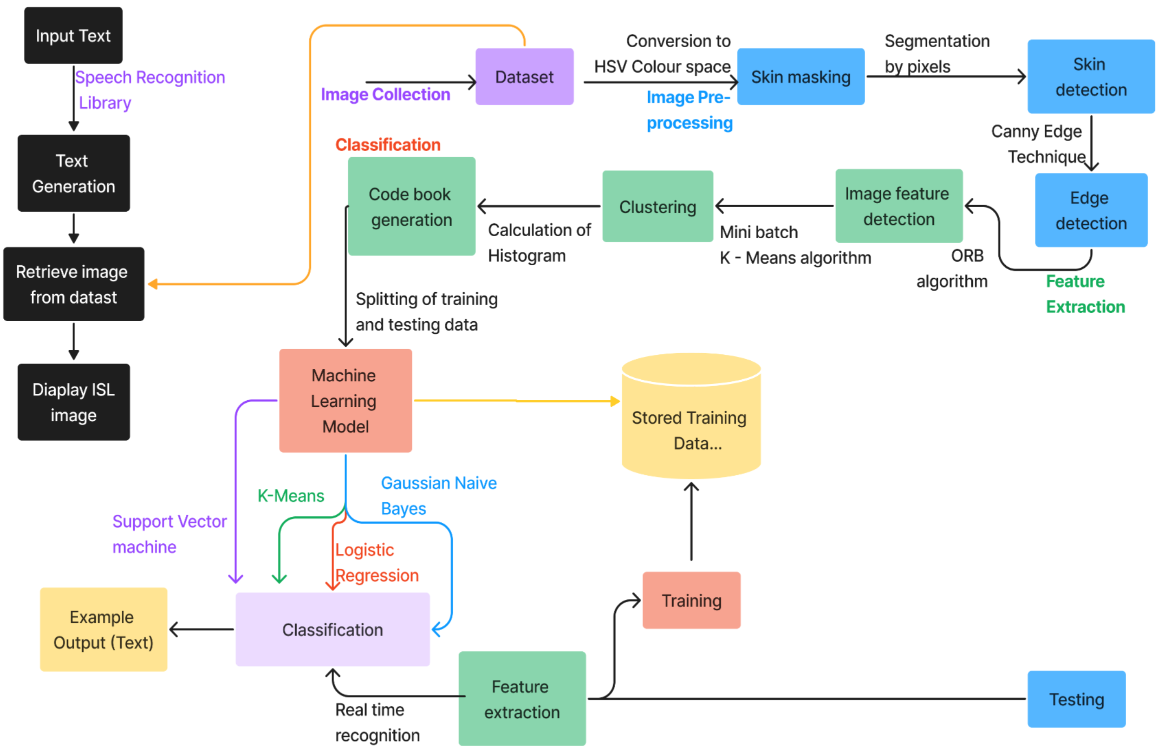
CHAPTER 3

METHODOLOGY

3.2 INTRODUCTION

This chapter provides a detailed description of the working of the sign language detection system which includes an elaborated approach of the methods that are implemented and various software tools used while obtaining the dataset, preprocessing, visualisation, building the machine learning model, training, and finally making the GUI.

3.2 BLOCK DIAGRAM



(Fig 3.1: Block diagram)

To obtain accurate predictions for input symbols and the output text, various classification techniques are implemented. Firstly, image collection is performed by importing a dataset from a credible source like Kaggle.
Image pre-processing is performed on each iteration of the input symbols through consecutive steps of skin masking, skin detection and edge detection respectively. The raw images are converted to HSV colour space for skin masking. After that skin detection is done by segmenting each pixel. Edge Detection is done by implementing Canny Edge Detection Technique.
Feature Extraction is done on the pre-processed images. ORB (Oriented FAST and Rotated BRIEF) algorithm is used for image feature detection. There are other more sophisticated algorithms to perform the task of feature detection in images such as SURF, SIFT, etc. However, we chose the efficient method with respect to the task to be performed.
Further, we perform the Clustering of Images using the Mini-Batch K-Means algorithm. The visual words are used for generating histograms by Code book generation. The histograms representing the different feature vectors are then utilised for clustering.
This is followed by four types of classification, namely, Support Vector Machine, K-Means , Logistic Regression and Gaussian Naive Bayes and their respective evaluation.

3.3 COLLECTION OF IMAGE DATASET

It is a very crucial part of the research works in all the arenas as it is fundamental to foster the development of any machine or deep learning model. However, it is full of challenges. During data collection, the biggest challenge we faced was that there were no standard datasets for Indian sign language available. Therefore, as part of this project, we attempted to manually construct a dataset that could help us overcome this problem.

First of all, we captured the videos using a webcam where various signs were taken into account. 26 different alphabets (A-Z) and 10 numeric signs (0–9) were considered from 3 persons. For the quality of the pictures and elimination of the background noises, the position of the camera is very critical. To add variations in the dataset, two options were used for capturing the images. The first one is the default method, which performs the skin segmentation on the image and can be used with a plain colour background.

In the second method, we have used the concept of running averages, in which some of the initial frames are considered as background and any new object after the initial frames is considered as foreground, thereby making the extraction process easier. The dataset was created by taking into account both of these approaches in order for the model to perform well in diverse scenarios.
The signs obtained from the live video were converted into frames, which were further extracted using a pixel value threshold. The produced frames had a resolution of 250*250 so that less computational power is required for pre-processing. Each sign folder contained around 1000 images of each sign. Hence the total number of images in the dataset were 36,000 for both image acquisition methods. The signs involved the use of a single hand as

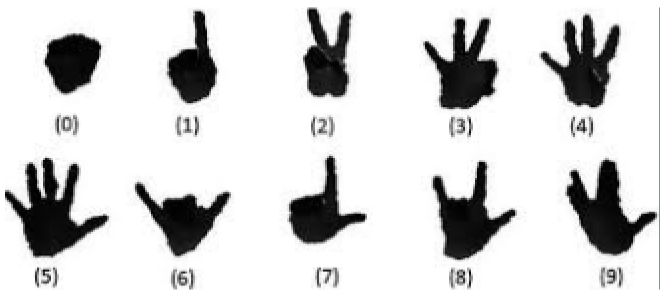
well as of both hands. The images were captured in different rotations and stored in grayscale format with. jpg extension.

(Fig 3.2: Dataset)

3.4 IMAGE PRE-PROCESSING

3.4.1 Pre processing

The image is made ready for feature detection and extraction in this phase. To preserve uniformity of scale, the dimensions of all the images are kept the same. In the default option, the captured video frame is converted into HSV colour space for the images acquired with the plain background. As the hue colour of the skin is different from that of the background, it gets extracted easily. An experimental threshold is then applied to the frame that calculates hue and filters out the skin coloured pixels from the image. Further, the image is binarized, blurring is done to remove noises and maximum contour is obtained from the result assuming that the contour with the largest area represents the hands. Errors are further removed by applying the median filter and morphological operations. In the second method for the images with the running background, the first 30 frames are considered as background and for the remaining frames the absolute difference is calculated between the additive sum of those 30 frames and the new frame, which gives us the foreground region of the current frame. The images are first converted into grayscale and then a Gaussian filter is applied. For hand segmentation, a mask is created by extracting the maximum connected region in the foreground assuming it to be the hand. Noises are further removed by applying morphological operations like erosion and dilation. After this, the canny function is used in which the gradient of each pixel calculates the edge strength and direction of the images. Compared to the original image, this results in a shift of intensity and the edge is detected easily. The pre-processed images from both the options are shuffled to add variation in the dataset.



(Fig 3.3: Thresholding of dataset images)

3.5. FEATURE EXTRACTION

(Fig 3.4 :Example of ORB process for feature extraction)

This phase involves building a Bag of Visual Words (BOVW) which includes feature extraction, clustering of features, codebook construc- tion for the model, and generation of histograms. The Bag of Visual Words (BOVW) is a widely used image classifica- tion model whose definition is adapted from data retrieval and NLP'S (Natural Language Processing) Bag of Words (BOW)

In this, we count the number of times each word appears in a text, use each word's frequency to get the keywords and produce a histogram of frequency from it. This idea is changed in such a way that instead of words, we use the image features as words. To construct a vocabulary where image is represented as a frequency histogram of characteristics obtained, the image descriptors and key points are used. Later on, the category of another comparable image can be predicted from this frequency histogram. As discussed, the first step in building a bag of visual words (BOVW) is to extract descriptors from each image in the dataset. The descriptor is a 64-member vector for each interest point in the execution used which defines the distribution of the intensity material within the neighborhood of the interest point.

For this, we use ORB (Oriented FAST and Rotated BRIEF) as a local feature detector and descriptor instead of SURF. ORB features are also robust against rotation, variance, and occlusion, and are faster to compute than SURF features.

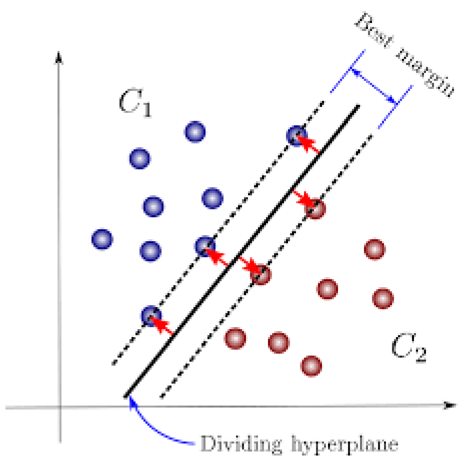
An image is represented as a set of image descriptors given by ORB as Eq (1). Im = {d1 , d2 , d3 , ...dn } (1), where di is the color, shape, etc. of the hands, and n denotes the total image descriptors. To generate a codebook, we cluster all the features obtained after applying ORB. This is done to group similar features so that it is possible to use the core and cluster them as the dictionary's visual keyword. We use the K-means algorithm to perform clustering, with a value of k as 180 for this purpose. For codebook generation, the resulting cluster centres (i.e., centroids) are treated as our code vectors. A codebook is used for quantizing features where it takes a feature vector as input and maps it to the index of the nearest code vector. The constructed vocabulary can be represented as: v = {w1, w2 , w3, ...wk } (2), where k is the total number of clusters, i.e. 180.

The mapping of each descriptor to the nearest visual word is done according to the Eq (3). w(d) = arg min Dist(w, d) (3), where w (di) depicts the visual word assigned to the ith descriptor and Dist (w,di) represents the distance between the visual word w and descriptor di. The last step is the generation of histograms for all the images which is done by calculating the frequency of occurrence of each visual word in an image. The count of bins in the histogram is equal to the total number of visual words in the dictionary i.e. k and is represented by Eq (4). bini = C(Di), where Di is the set of all the descriptors corresponding to a particular visual word wi in the image and C (Di) is the cardinality representing the count of the elements in set Di. For every visual word in the image, this is repeated to obtain final histograms that are then passed for recognition to the classifier along with their respective labels.

3.6. CLASSIFICATION

3.6.1. SUPPORT VECTOR MACHINE (SVM)

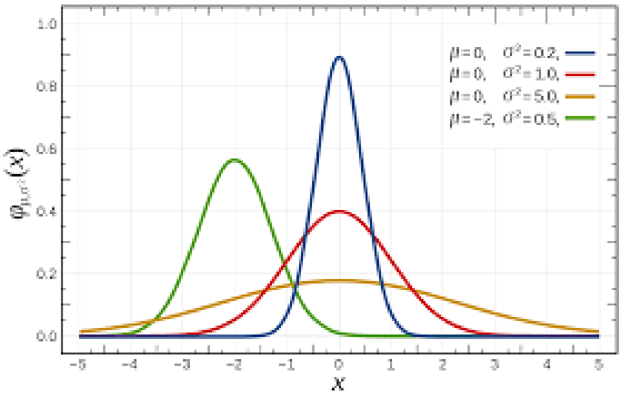
The Support Vector Machine (SVM) is a supervised model that can solve both linear and non-linear problems for classification and regres- sion problems. It operates on the idea of decision planes that specify boundaries for decisions. For this classification, we have used SVM with a linear kernel. We have passed the histograms of visual words to the SVM as feature vectors for the classification and recognition of ISL signs. The training is done using a total of 28,800 images. After the training is completed, the performance of the classifier is checked on the testing set which has a total of 7236 images, and its performance is evaluated on various parameters like accuracy, precision, recall, etc



(Fig 3.5 : Support Vector Machine algorithm)

3.6.2. GAUSSIAN NAIVE BAYES CLASSIFIER

The Gaussian Naive Bayes Classifier is a simple yet effective probabilistic classifier used in machine learning. It is based on Bayes' theorem and assumes that the features are independent of each other. It is called "naive" because it makes a simplifying assumption that the features are independent of each other, which is often not the case in real-world applications. The classifier works by calculating the conditional probability of each class given the features, and then selecting the class with the highest probability as the predicted class. The conditional probability is calculated using Bayes' theorem, which states that the probability of a hypothesis (in this case, a class) given the evidence (the features) is proportional to the probability of the evidence given the hypothesis multiplied by the prior probability of the hypothesis.



(Fig 3.6 : Gaussian Naive Bayes classifier algorithm)

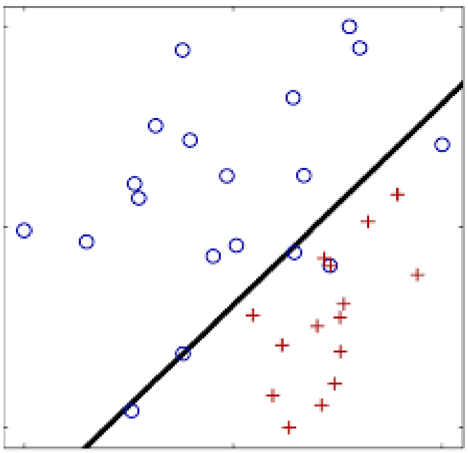
3.6.3. LINEAR REGRESSION

Linear Regression is a popular and widely used technique in machine learning for predicting a continuous output variable (also called the dependent variable) from one or more input variables (also called independent or predictor variables). It is a supervised learning algorithm, which means it requires labelled training data to learn the relationship between the input variables and the output variable. The basic idea behind linear regression is to find a linear relationship between the input variables and the output variable. This is achieved by fitting a linear equation to the data, where the coefficients of the equation are learned from the training data.

The general form of a linear regression equation is:

y = b0 + b1x1 + b2x2 + ... + bn*xn

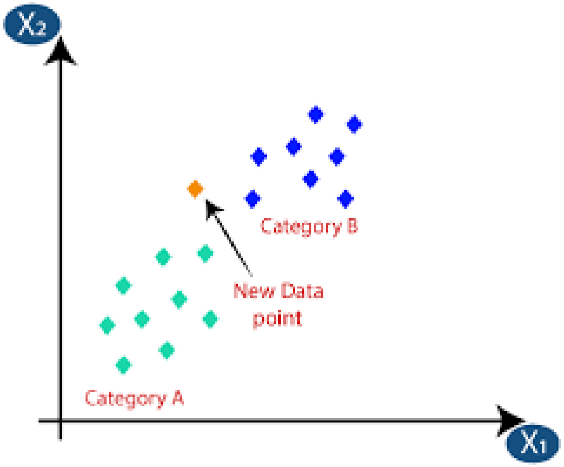
where y is the output variable, x1, x2, ..., xn are the input variables, b0 is the intercept or bias term, and b1, b2, ..., bn are the coefficients or weights that determine the effect of each input variable on the output variable.



(Fig 3.7: Logistic Regression algorithm)

3.6.4. K-NEAREST NEIGHBOURS

K-Nearest Neighbors (KNN) is a non-parametric machine learning algorithm used for classification and regression. It is a simple yet effective algorithm that works by finding the k closest training examples in the feature space and using their labels (in classification) or values (in regression) to predict the label or value of a new example. The algorithm involves three main steps: (1) computing the distance between the new example and each training example, (2) selecting the k training examples with the smallest distance, and (3) using their labels or values to predict the label or value of the new example.



(Fig3.8 : K-Nearest Neighbor algorithm)

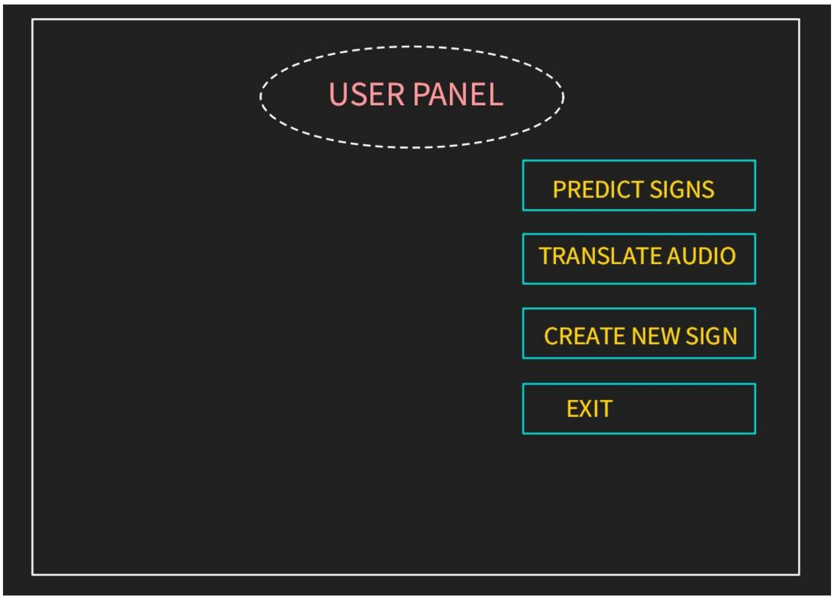
3.7. REVERSE RECOGNITION

The reverse process is essential in a sign language recognition system to provide a dual mode of communication between the speech impaired and hearing majority [28]. We have implemented this mode of communication in our system. Here text (English alphabets) is given as the input in the form of speech by the user, where it is mapped onto the labels and corresponding signs (images stored in database) are displayed to the user in a sequence. The speech recognition is done using the Google Speech API.

3.8 GUI IMPLEMENTATION

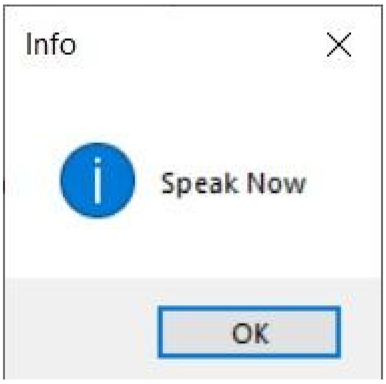
We connect the GUI with the machine learning model for attaining the input and output operations via the user. It is also used for the implementation of the Reverse Speech-to-text recognition task.

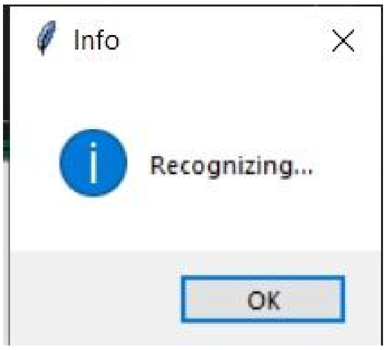
3.8.1 USER PANEL



(Fig 3.9: User Panel)

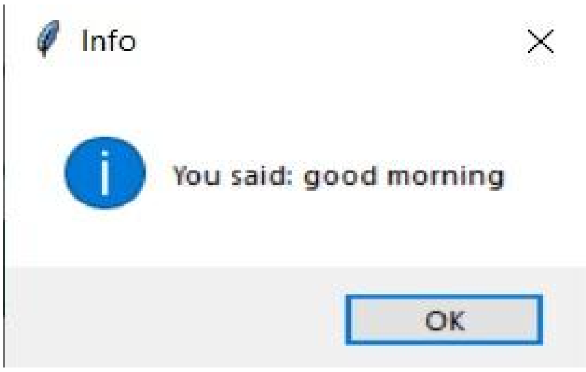
3.8.2 SPEAK/RECOGNIZING DIALOG BOX





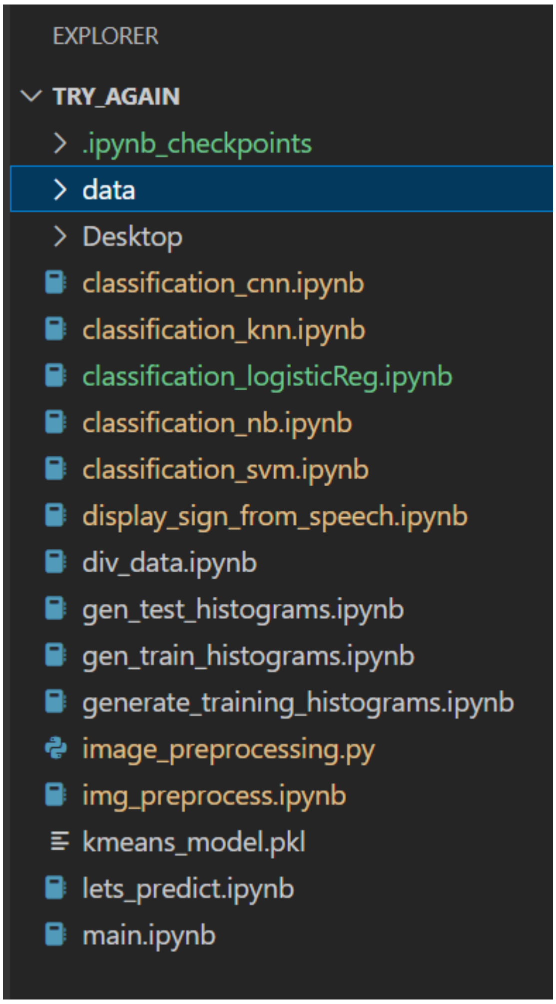
(Fig 3.10 & Fig 3.11: Dialog Boxes)

3.8.3 DISPLAY OUTPUT RECOGNIZED TEXT



(Fig 3.12: Output of recognized text)

3.9 FILE STRUCTURE



(Fig 3.13: File Structure)

- Image_preprocessing.py performs the tasks of image vectorization, thresholding, edge detection and other image processing tasks on the input images. And div_data.ipynb is then used to segment the data into training and testing datasets in a 60-40, 70-30 split. The splitting ratio for train and test can be determined by the user.
- Generate_training-histograms.ipynb performs the task of clustering in an unsupervised manner using the relevant K-means mini-batch clustering algorithm. It generates the bag of visual word model, and clusters similar features together. The BOVW model is used to generate the training histograms that are then fed into the machine learning model for classification and prediction.
- Classification_svm.ipynb, etc. are classification algorithm based functions which can each be used to implement classification on the images, and evaluate the results.

3.10 TOOLS USED

For the project, Python 3 programming language has been used due to the availability of powerful Machine Learning and Computer Vision libraries, as well as feasibility of deployment. Python 3 is today the most widely and powerfully used programming language for machine learning applications.

- opencv-python
- opencv-contrib-python
- numpy
- Panda
- sklearn

3.10.1 OPENCV - PYTHON

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimised library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

3.10.2 OPENCV-CONTRIB-PYTHON

OpenCV contrib is a specialised module present in the Python programming language, which is exclusively needed for the system to run SURF feature descriptions alongside the OpenCV module present in the open-source library. It must be noted that the algorithm is subjected to encryption with copyrights, and hence it should only be used for learning and development processes or personal use and should not be used for any production purposes unless the copyright status of the module is completely understood and addressed by the user. OpenCV provides pre-compiled Java jars and all files on their website.

However, if you use the pre-bundled OpenCV jar files and DLL files, you will get errors when working through the programs that are developed around the OpenCV tutorials. This is especially seen when dealing with the org.feature packages that will not be found and will prevent the quote from being compiled.

3.10.3 NUMPY

NumPy (Numerical Python) is an open source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures (you'll find more information about this in later sections). It provides an array, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

3.10.4 PANDAS

Pandas is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

3.10.5 SKLEARN

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistent interface in

Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

CHAPTER 4

RESULT ANALYSIS

4.1 INTRODUCTION

In this chapter, the analysis of the results and ramifications of the findings of the project i.e. Indian Sign Language prediction system using Machine Learning will be discussed in detail.

4.2 RESULT ANALYSIS

4.2.1 Analysis of Gaussian Naive Bayes Algorithm on the dataset:

<u>ITERATIONS:</u>

Here we have generated a comprehensive comparison for the parameters of Precision,Recall and F1 score, for Gaussian Naive Bayes algorithms for the predefined dataset.

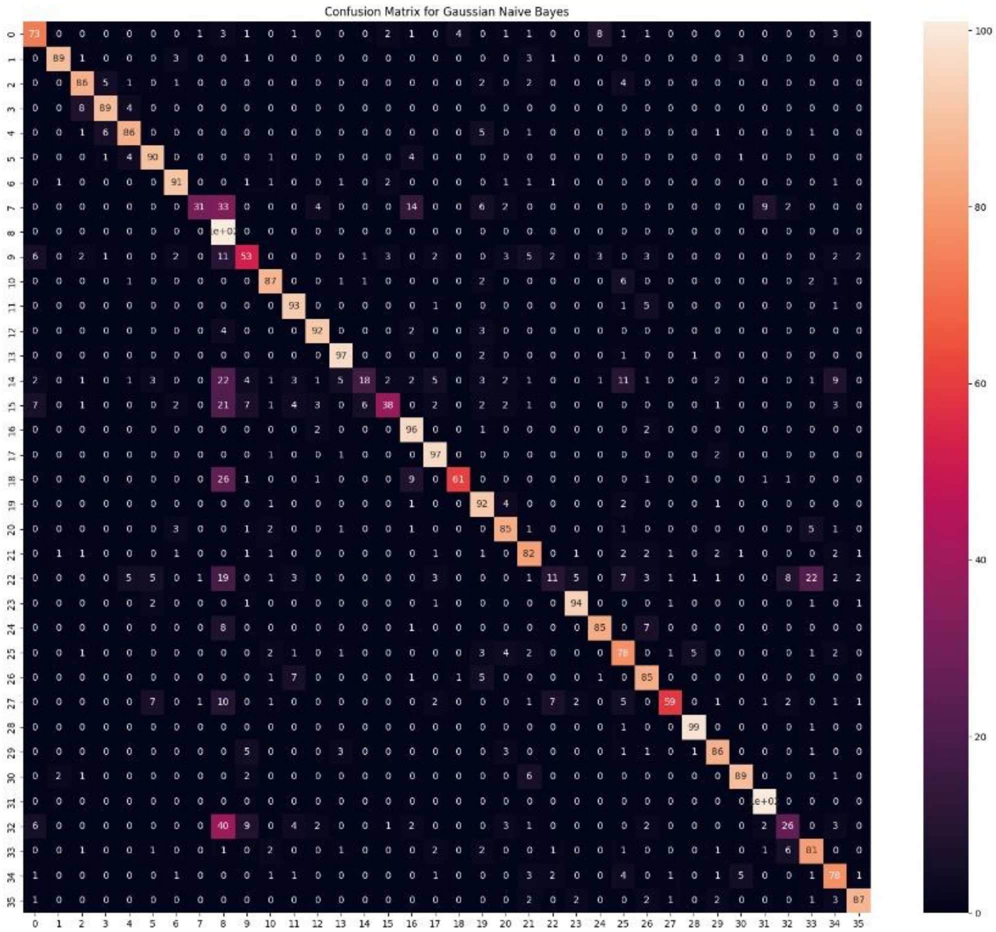
Naive Bayes Finished !
0.7662266226622663

	precision	recall	f1-score	support
0	0.76	0.72	0.74	101
1	0.96	0.88	0.92	101
2	0.83	0.85	0.84	101
3	0.87	0.88	0.88	101
4	0.84	0.85	0.85	101
5	0.83	0.89	0.86	101
6	0.88	0.90	0.89	101
7	0.91	0.31	0.46	101
8	0.34	1.00	0.51	101
9	0.61	0.52	0.56	101
10	0.84	0.86	0.85	101
11	0.79	0.92	0.85	101
12	0.88	0.91	0.89	101
13	0.87	0.96	0.92	101
14	0.69	0.18	0.28	101
15	0.79	0.38	0.51	101
16	0.72	0.95	0.82	101
17	0.83	0.96	0.89	101
18	0.92	0.60	0.73	101
19	0.71	0.91	0.80	101
20	0.77	0.84	0.81	101
21	0.72	0.81	0.76	101
22	0.44	0.11	0.17	101
23	0.90	0.93	0.92	101
24	0.87	0.84	0.85	101
25	0.62	0.77	0.69	101
26	0.74	0.84	0.79	101
27	0.91	0.58	0.71	101
28	0.93	0.98	0.95	101
29	0.85	0.85	0.85	101
30	0.90	0.88	0.89	101
31	0.88	1.00	0.94	101
32	0.58	0.26	0.36	101
33	0.69	0.80	0.74	101
34	0.69	0.77	0.73	101
35	0.92	0.86	0.89	101
accuracy			0.77	3636
macro avg	0.79	0.77	0.75	3636
weighted avg	0.79	0.77	0.75	3636

(Fig 4.1: table of iterations for GNB algorithm)

CONFUSION MATRIX:

A confusion matrix can be used to perfectly analyze the potential of a classifier. Given below is the confusion matrix generated for the Gaussian Naive Bayes algorithm applied on the predefined dataset. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to zero. But here we can see that is not the case, we have several non-diagonal elements that are giving non-zero values, hence we can conclude that this is not the best classifier algorithm for the given dataset.



(Fig 4.2 : Confusion matrix for GNB algorithm)

4.2.2 Analysis of Support Vector Machine Algorithm on the dataset:

ITERATIONS:

Here we have generated a comprehensive comparison for the parameters of Precision,Recall and F1 score, for Support Vector Machine algorithm.

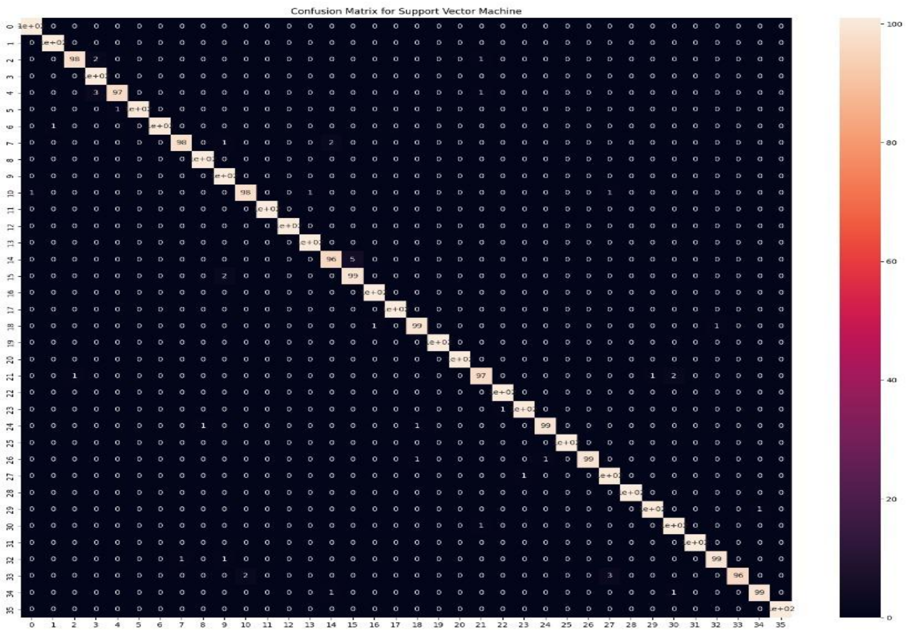
0.9876237623762376

	precision	recall	f1-score	support
0	0.99	1.00	1.00	101
1	0.99	1.00	1.00	101
2	0.99	0.97	0.98	101
3	0.95	1.00	0.98	101
4	0.99	0.96	0.97	101
5	1.00	0.99	1.00	101
6	1.00	0.99	1.00	101
7	0.99	0.97	0.98	101
8	0.99	1.00	1.00	101
9	0.96	1.00	0.98	101
10	0.98	0.97	0.98	101
11	1.00	1.00	1.00	101
12	1.00	1.00	1.00	101
13	0.99	1.00	1.00	101
14	0.97	0.95	0.96	101
15	0.95	0.98	0.97	101
16	0.99	1.00	1.00	101
17	1.00	1.00	1.00	101
18	0.98	0.98	0.98	101
19	1.00	1.00	1.00	101
20	1.00	1.00	1.00	101
21	0.97	0.96	0.97	101
22	0.99	1.00	1.00	101
23	0.99	0.99	0.99	101
24	0.99	0.98	0.99	101
25	1.00	1.00	1.00	101
26	1.00	0.98	0.99	101
27	0.96	0.99	0.98	101
28	1.00	1.00	1.00	101
29	0.99	0.99	0.99	101
30	0.97	0.99	0.98	101
31	1.00	1.00	1.00	101
32	0.99	0.98	0.99	101
33	1.00	0.95	0.97	101
34	0.99	0.98	0.99	101
35	1.00	1.00	1.00	101
accuracy			0.99	3636
macro avg	0.99	0.99	0.99	3636
weighted avg	0.99	0.99	0.99	3636

(Fig 4.3 : table of iterations for SVM algorithm)

CONFUSION MATRIX:

Given below is the confusion matrix generated for the Support Vector Machine algorithm applied on the predefined dataset. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to zero. As we can see here, no non-diagonal elements have non-zero values, and only diagonal elements have non-zero values. Thus, we can conclude that this classification algorithm works best for this dataset.



(Fig 4.4: Confusion matrix for SVM algorithm)

4.2.1 Analysis of Logistic Regression Algorithm on the dataset:

ITERATIONS:

Here we have generated a comprehensive comparison for the parameters of Precision,Recall and F1 score, for Logistic regression algorithm.

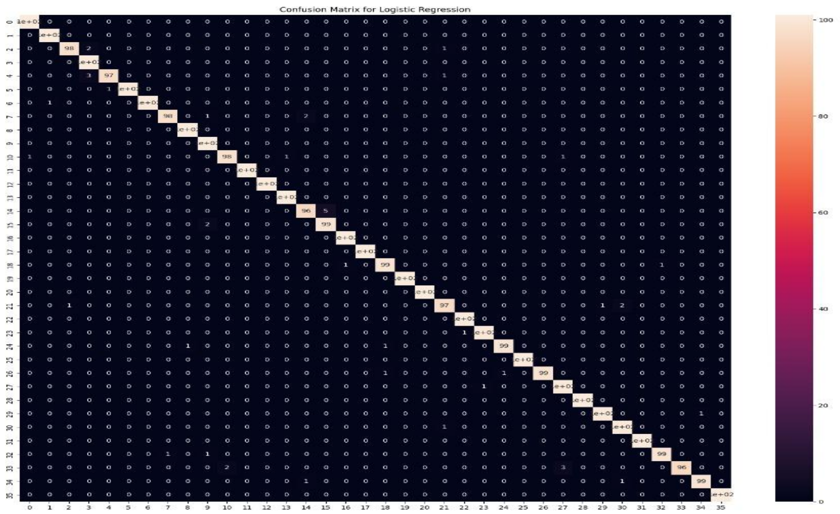
0.985973597359736

	precision	recall	f1-score	support
0	0.99	1.00	1.00	101
1	1.00	0.99	1.00	101
2	0.97	0.98	0.98	101
3	0.97	1.00	0.99	101
4	0.98	0.98	0.98	101
5	0.99	0.98	0.99	101
6	0.99	0.98	0.99	101
7	1.00	0.99	1.00	101
8	0.98	0.98	0.98	101
9	0.97	1.00	0.99	101
10	1.00	1.00	1.00	101
11	1.00	0.99	1.00	101
12	0.98	0.96	0.97	101
13	1.00	1.00	1.00	101
14	0.94	0.98	0.96	101
15	0.97	0.98	0.98	101
16	0.99	1.00	1.00	101
17	1.00	1.00	1.00	101
18	0.97	0.96	0.97	101
19	1.00	1.00	1.00	101
20	0.99	1.00	1.00	101
21	0.98	0.92	0.95	101
22	0.99	1.00	1.00	101
23	0.99	1.00	1.00	101
24	0.99	0.97	0.98	101
25	0.99	0.99	0.99	101
26	1.00	1.00	1.00	101
27	0.98	0.98	0.98	101
28	1.00	0.99	1.00	101
29	0.97	0.99	0.98	101
30	0.96	0.99	0.98	101
31	0.99	1.00	1.00	101
32	0.99	1.00	1.00	101
33	1.00	0.97	0.98	101
34	0.98	0.94	0.96	101
35	1.00	1.00	1.00	101
accuracy			0.99	3636
macro avg	0.99	0.99	0.99	3636
weighted avg	0.99	0.99	0.99	3636

(Fig 4.5 : Iterations for Logistic Regression algorithm)

<u>CONFUSION MATRIX:</u>

Given below is the confusion matrix generated for the Logistic regression algorithm applied on the predefined dataset. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to zero. As we can see here, no non-diagonal elements have non-zero values, and only diagonal elements have non-zero values. Thus, we can conclude that this classification algorithm works best for this dataset.



(Fig 4.6: Confusion matrix for Logistic regression algorithm)

4.2.1 Analysis of K-Nearest Neighbour Algorithm on the dataset:

<u>ITERATIONS:</u>

Here we have generated a comprehensive comparison for the parameters of Precision,Recall and F1 score, for the K-Nearest Neighbour algorithm.

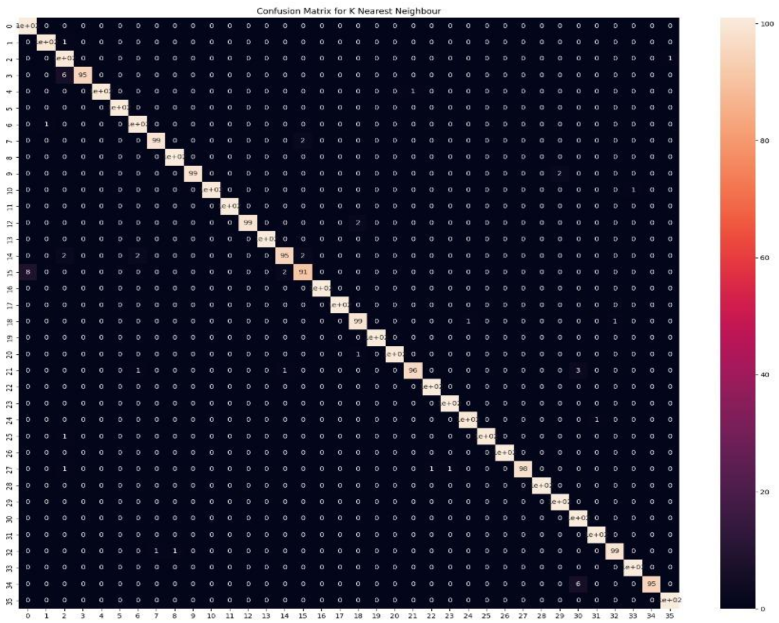
0.9851485148514851

	precision	recall	f1-score	support
0	0.93	1.00	0.96	101
1	0.99	0.99	0.99	101
2	0.90	0.99	0.94	101
3	1.00	0.94	0.97	101
4	1.00	0.99	1.00	101
5	1.00	1.00	1.00	101
6	0.97	0.99	0.98	101
7	0.99	0.98	0.99	101
8	0.99	1.00	1.00	101
9	1.00	0.98	0.99	101
10	1.00	0.99	1.00	101
11	1.00	1.00	1.00	101
12	1.00	0.98	0.99	101
13	1.00	1.00	1.00	101
14	0.97	0.94	0.95	101
15	0.96	0.90	0.93	101
16	1.00	1.00	1.00	101
17	1.00	1.00	1.00	101
18	0.97	0.98	0.98	101
19	1.00	1.00	1.00	101
20	1.00	0.99	1.00	101
21	0.99	0.95	0.97	101
22	0.99	1.00	1.00	101
23	0.99	1.00	1.00	101
24	0.99	0.99	0.99	101
25	1.00	0.99	1.00	101
26	1.00	1.00	1.00	101
27	1.00	0.97	0.98	101
28	1.00	1.00	1.00	101
29	0.98	1.00	0.99	101
30	0.92	1.00	0.96	101
31	0.99	1.00	1.00	101
32	0.99	0.98	0.99	101
33	0.99	1.00	1.00	101
34	1.00	0.94	0.97	101
35	0.99	1.00	1.00	101
accuracy			0.99	3636
macro avg	0.99	0.99	0.99	3636
weighted avg	0.99	0.99	0.99	3636

(Fig 4.7: Iterations for KNN algorithm)

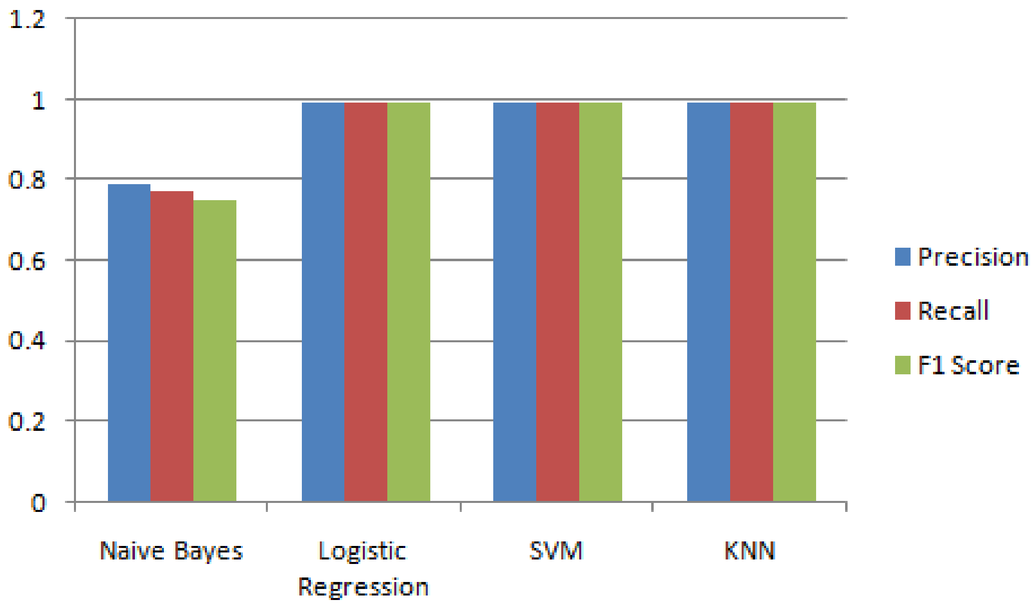
CONFUSION MATRIX:

Given below is the confusion matrix generated for the K-Nearest Neighbor algorithm applied on the predefined dataset. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. Hence, the best classifier will have a confusion matrix with only diagonal elements and the rest of the elements set to zero. As we can see here, no non-diagonal elements have non-zero values, and only diagonal elements have non-zero values. Thus, we can conclude that this classification algorithm works best for this dataset.



(Fig 4.8: Confusion matrix for KNN algorithm)

4.3 GRAPHICAL COMPARISON OF THE RESULTS OBTAINED FROM DIFFERENT ALGORITHMS ON THE DATASET:



(Fig 4.9: Graphical comparison between various classification algorithms*)*

4.3 RESULT OUTCOMES

We implemented image classification using four different supervised classification algorithms and we found that for this task Naives Bayes Classifier gives the least optimum results. This can be attributed to the following reasons:

- Naive Bayes assumes independence of features: It assumes that all features are independent, which may not hold true for sign language image prediction. Features such as the position of the hand and fingers may be highly dependent on each other, and therefore Naive Bayes may not be the best algorithm to capture such complex relationships.
- Limited expressive power: Naive Bayes has limited expressive power compared to other algorithms, which can limit its ability to capture complex patterns and relationships in the data. For example, KNN can capture more complex decision boundaries compared to Naive Bayes.
- Sensitivity to imbalanced data: Naive Bayes can be sensitive to imbalanced data, where one class has significantly more samples than the other. This can be a problem for sign language image prediction, where some signs may be more common than others.

The classification algorithms K-Nearest Neighbors, Support Vector Machine and Logistic Regression may perform better because they can capture complex relationships between features. They have higher expressive power which makes a better model of the data. Moreover, they can handle imbalanced data more robustly. However, the performance of these algorithms can also depend on the specific dataset and features used, and it's always a good idea to experiment with multiple algorithms to determine the best one for a particular task.

CHAPTER 5:

CONCLUSION AND FUTURE SCOPE OF WORK

5.1 INTRODUCTION

In this chapter, we discuss the conclusive findings and accomplishments of our project work and the model developed herein. We also look at future scope of work in the field, including applications and further scope for development.

5.2 BRIEF SUMMARY OF THE WORK

- A computer vision system to help the deaf and mute communicate in a shopping setting, with text-to-speech integration using python libraries. The goal of this system is to provide a way for deaf and mute individuals to communicate with sales associates in a shopping setting, where there may be a need for assistance or inquiries. The system consists of a camera-based computer vision module that can detect and recognize sign language gestures made by the user. Once a gesture is detected, the system converts it to text and uses text-to-speech conversion to communicate with the sales associate.
    - For example, if a deaf person at the shopping mall wants to seek some suggestions for clothing then and if there are no ISL-speaking people around then one might not get the same seamless shopping experience like the one with the ability to speak English.
- A system to train and raise awareness about Indian Sign Language amongst the general masses can be developed via this project. This system could be used in different settings, such as in schools for the hearing-impaired or for community training programs. Additionally, it could also be used as a tool for raising awareness about ISL among the general population by showcasing the benefits of learning ISL and the importance of inclusivity and accessibility for hearing and speech-impaired individuals. Such a system would not only make it easier for people to learn and understand ISL, but it could also help promote a more inclusive and accessible society.

Developing a system that helps in the translation of Indian Sign Language to other languages besides English for effective communication. This system can also be useful in different settings such as education, healthcare, and public services, where it can help to improve the communication between the deaf community and service providers. It can also help to raise awareness about Indian Sign Language and promote its use among the general masses. By developing a sign language translation system, we can take a step towards creating a more inclusive society where everyone can communicate effectively regardless of their ability to hear or speak.

5.3 FUTURE SCOPE OF WORK

- Skeleton-based Action Recognition with Convolutional Neural Networks' can be integrated with it for better understanding, along with facial expression recognition, as communication is not only words, but gestures and expressions.
- Web-services can use it as an extension to facilitate communication between deaf/mute people and normal people.
- In our project currently we have only taken 'Fingerspelling' into account, in the recent future we can scale our project to also include 'Word Recognition' and 'Gesture/Facial Expression Recognition' too.

REFERENCES

[1]. "Indian Sign Language Recognition Using Convolutional Neural Networks," by A. K. Singh, A. K. Tiwari, and D. D. Doye, International Conference on Communication, Computing and Networking (ICCCN), 2018.

[2]. "A Survey on Indian Sign Language Recognition," by A. H. Shaikh and S. S. Pawar, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2017.

[3]. "Real-time Sign Language Recognition Using Deep Learning," by S. Shah and M. Shah, International Conference on Advances in Computing, Communication and Control (ICAC3), 2019.

[4]. "Indian Sign Language Recognition Using Machine Learning Techniques: A Review," by N. N. Hengade and M. D. Kokare, International Conference on Computing, Communication and Automation (ICCCA), 2017.

[5]. "Indian Sign Language Recognition Using Leap Motion Controller," by S. P. Patil and S. K. Patil, International Conference on Computing, Communication and Automation (ICCCA), 2017.

[6]. Yu-Gang Jiang, Jun Yang, Chong-Wah Ngo, and Alexander Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. Multimedia, IEEE Transactions on, 12:42 – 53, 02 2010.

[7]. David G. Lowe. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 60(2):91–110, November 2004.

[8]. Dutta, Kusumika& K, Satheesh& S, Anil & Sunny, Breeze. (2015). Double handed Indian Sign Language to speech and text. 374- 377. 10.1109/ICIIP.2015.7414799.

[9] Mukesh Kumar Makwana, " Sign Language Recognition", M.Tech thesis, Indian Institute of Science, Bangalore

[10] Pigou, Lionel, et al. ”Sign language recognition using convolutional neural networks.” Workshop at the European Conference on Computer Vision. Springer International Publishing, 2014.

[11] Escalera, Sergio, et al. ”Chalearn looking at people challenge 2014: Dataset and results.” Workshop at the European Conference on Computer Vision. Springer International Publishing, 2014.

[12] Kuznetsova Alina, Laura Leal-Taix, and Bodo Rosenhahn. ”Real-time sign language recognition using a consumer depth camera.” Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.

[13] J. -. Lementec and P. Bajcsy, "Recognition of arm gestures using multiple orientation sensors: gesture classification", Proceedings. The 7 th International IEEE

Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749), Washington, WA, USA, 2004, pp. 965-970.doi: 10.1109/ITSC.2004.1399037.

[14] S. Hussain, R. Saxena, X. Han, J. A. Khan, and H. Shin, “Hand gesture recognition using deep learning”, 2017 International SoC Design Conference (ISOCC) , Seoul, pp. 1-6.

[15] T. Yamashita and T. Watasue, "Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning", 2014 IEEE International Conference on Image Processing (ICIP) , Paris, 2014, pp. 853-857.

[16] Pei Xum “A real time hand gesture recognition and human computer interaction,” Dept. of Electrical and Computer Engineering, University of Minnesota, 2017, pp. 1-8.

[17] B. Liao, J. Li, Z. Ju and G. Ouyang, "Hand Gesture Recognition with Generalized Hough Transform and DC-CNN Using Realsense," 2018 Eighth International Conference on Information Science and Technology (ICIST) , Cordoba, 2018, pp. 84-90.

[18] "Indian Sign Language Recognition Using Hand Gesture Recognition Techniques," by S. Jadhav and P. More, International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 2016.

[19] "Indian Sign Language Recognition System Using Principal Component Analysis and Neural Networks," by S. A. Mahajan and V. M. Thakare, International Journal of Computer Science and Information Technologies (IJCSIT), 2017.

[20]"Indian Sign Language Recognition Using Artificial Bee Colony Algorithm and Multilayer Perceptron," by P. R. Somani, K. R. Kulkarni, and R. K. Gupta, International Conference on Signal Processing and Communication (ICSC), 2016.