Acknowledgements

I want to express my gratitude and admiration to **Dr. Soumya Ranjan Mishra** for his great leadership and unwavering support throughout the full development of my project, Sign Bridge. I was able to surpass my own objectives and reach my goals thanks to Dr. Soumya Ranjan Mishra's excellent counsel and unwavering support throughout the entire project.

Dr. Soumya Ranjan Mishra served as an important mentor to me throughout the project, offering me priceless insights, techniques, and resources to aid me in navigating the difficulties and intricacies of my job. I could not have asked for a more committed or knowledgeable tutor to help me through this process; his in-depth knowledge and expertise in the industry were genuinely priceless.

Dr. Soumya Ranjan Mishra's enthusiasm and passion for the subject were contagious and motivating, and his continuous support and encouragement kept me inspired and on task despite obstacles or failures. I am sincerely appreciative of the chance to learn from such an exceptional teacher, and I am confident that the abilities and information I have developed will be helpful to me in all of my future endeavours.

I'd like to end by expressing my sincere gratitude to Dr. Soumya Ranjan Mishra for his professional advice, perseverance, and support during the Sign Bridge project. I count it a privilege to have had the chance to work with a mentor of his calibre, and I have no doubt that his contributions have been essential to the success of my work.

SIDDARTH DUTTA

TUSHAL JAIN

ADITYA SRIVASTAV

ASHISH KUMAR MISHRA

ABSTRACT

With the help of a combination of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models, this study intends to create a system for the recognition of sign language. The suggested method aims to improve communication between deaf or hard-of-hearing people by converting their sign language motions into spoken or written language. The CNN model is used to extract spatial features from the input data, while the LSTM model is utilised to capture the sequential aspect of sign language. The task at hand is gathering and preprocessing a dataset of sign language movements, training the LSTM and CNN models on it, then assessing the system's performance on a different test set. The findings of this study may guide the creation of future systems that are more precise and effective as well as help determine how well LSTM and CNN models perform as tools for sign language recognition. The project's five focal elements are deep learning, CNN, LSTM, sign language recognition, and gesture recognition.

1. Sign language recognition : The main goal of this research is to create a system that can accurately recognise and decipher sign language motions. The system will be created to improve communication for those who are hard of hearing or deaf.

2. LSTM and CNN models : Convolutional neural network (CNN) and long short-term memory (LSTM) models will be used in the project to recognise sign language. The sequential aspect of sign language will be captured using LSTM, and spatial information will be extracted from the input data using CNN.

3. Gesture recognition : The system's main focus will be on distinguishing distinct sign language motions, which serve as the basic units of sign language sentences. The system will be taught to appropriately translate these motions into spoken or written words.

4. Deep learning : The LSTM and CNN models will be trained on a sizable dataset of sign language motions using deep learning techniques. Preprocessing the data, training the models, and applying the right metrics to assess the system's success are all necessary steps in this process.

5. Real-time implementation : The creation of a real-time system that can be used to translate sign language gestures is one of the main goals of this research. To do this, the deep learning models must be accelerated and made more effective. They also need to be integrated with the proper hardware and software.

Contents

List of Figures

Chapter 1

Introduction

Millions of individuals use sign language, a visual language, to communicate on a global scale. For those who are deaf or hard of hearing, however, the absence of access to sign language interpretation might cause communication difficulties. The options for sign language recognition that are now on the market are limited and frequently call for specialised hardware or manual input from a sign language translator. The goal of this project is to use CNN and LSTM models to create a system for sign language recognition that is more accurate and effective.

Importance of the Project:

By offering real-time translation of sign language motions into written or spoken language, the suggested system has the potential to revolutionise communication in sign language. This will lower communication barriers and encourage inclusivity by allowing people who are deaf or hard of hearing to interact with others more effectively.

Current Gaps in Available Solutions:

Although there are several existing techniques for sign language recognition, their effectiveness and accuracy are frequently constrained. Many are less accessible to the general public because they need specialised equipment or manual input from a sign language interpreter. Additionally, more complicated sign language movements or sentences may be difficult for existing solutions to understand, creating significant communication hurdles.

Structure of the Report:

Using LSTM and CNN models, this paper will first give a thorough overview of the pertinent literature and prior research on the recognition of sign language. The procedures for gathering data, preparing it, and training and assessing the LSTM and CNN models will all be described in the methods section. The project's findings, including the precision and effectiveness of the suggested system, will be presented in the outcomes section. Finally, the analysis of the findings and suggestions for further study in deep learning models for sign language recognition are provided in the discussion and conclusion sections.

Chapter 2

Basic Concepts/ Literature Review

2.1 Machine Learning Concepts:

The suggested system for sign language recognition is based on deep learning principles of machine learning. Deep neural networks are used in the science of deep learning, a branch of machine learning, to identify intricate data patterns. In this study, LSTM and CNN models are used to recognise sign language while implementing deep learning principles. The models are used to classify new gestures in real-time and are trained using a collection of sign language movements. By applying supervised learning, the models are trained using the input data and their related labels. Metrics like accuracy and loss are used to assess the models' performance.

2.2 Deep Learning:

Deep neural networks are used in deep learning, a type of machine learning, to discover intricate patterns in data. Deep learning has demonstrated tremendous potential in a variety of applications, including speech recognition, image recognition, and natural language processing. This project will create a sign language recognition system that is precise, effective, and able to translate sign language motions in real-time using deep learning techniques like LSTM and CNN.

2.3 Sign Language Recognition:

The process of identifying and deciphering sign language movements using machine learning techniques is known as sign language recognition. Millions of people use sign language, a visual language, to communicate on a global scale, especially those who are deaf or hard of hearing. Systems that can translate sign language motions into written or spoken language in real time have the potential to revolutionise communication for these people.

2.4 Gesture Recognition:

The process of identifying and interpreting individual gestures, which serve as the basic units of sign language sentences, is known as gesture recognition. Machine learning methods like LSTM or CNN are frequently used by gesture recognition systems to analyse the spatial and temporal properties of the gestures and classify them into the appropriate categories.

2.5 Long Short-Term Memory (LSTM):

Recurrent neural networks (RNNs) of the LSTM variety are made to recognise long-term dependencies in sequential input. Memory cells and gates are used in LSTM networks to control the information flow into and out of the memory cells. Memory cells are used to store information over time. Because of this, LSTM is especially well suited for time series data applications like speech recognition and natural language processing.

2.6 MediaPipe:

An open-source cross-platform framework called MediaPipe is used to create multi-modal machine learning applications. In order to create a unique pipeline for processing several forms of input data, including video, audio, and sensor data, MediaPipe offers a variety of pre-built components. In this project, MediaPipe is employed to extract images from a video stream and prepare them for recognition of sign language.

2.7 OpenCV (cv2):

For processing and analysing visual data, OpenCV, an open-source computer vision toolkit, offers a wide range of tools and methods. Object detection, feature extraction, and image and video processing are all included in OpenCV. The images from the video stream are captured and preprocessed in this project using OpenCV (cv2).

2.8 Convolutional Neural Networks (CNN):

An image processing and analysis neural network called a CNN is used. Convolutional layers are used by CNN networks to extract spatial details from the input images, and pooling layers are then used to lower the dimensionality of the feature maps. Because of this, CNN is especially well-suited for applications involving visual data, like object or image detection.

Chapter 3

Problem Statement / Requirement Specifications

The challenge experienced by people with hearing impairments in communicating with people who do not understand sign language is the issue being addressed in this study. The idea is to create a system for sign language translation into spoken or written language that accurately recognises and decodes sign language motions. This will make it simpler and more effective for people with hearing loss to interact with others on a daily basis. The limited number of reliable options for sign language recognition currently on the market causes misunderstandings and frustration. Therefore, the requirement for a precise and trustworthy method of sign language recognition is essential.

3.1 Project Planning

The main objective of this research is to create a deep learning-based system for sign language recognition. The project will need to meet the following criteria:

1. Utilise MediaPipe and OpenCV to extract images from a video feed.
2. To enhance the photographs' quality and get rid of noise, preprocess them.
3. Create CNN and LSTM models to recognise sign language.
4. Using a dataset of sign language gestures, train the models.
5. Analyse the models' performance using metrics like accuracy and loss.
6. Create a system that recognises sign language in real time by including the models.

3.2 Project Analysis

The problem statement must be examined for any potential ambiguities or problems before beginning the creation of the sign language recognition system. Lack of a substantial and varied collection of sign language gestures is one potential problem that can impair the system's accuracy and dependability.

3.3 System Design

3.3.1 Design Constraints

To capture and prepare images for the sign language recognition system, MediaPipe and OpenCV will be used. The programme will be made to function on a typical personal computer equipped with a webcam.

3.3.2 System Architecture **OR** Block Diagram

Real-time sign language recognition, deep learning models, and picture acquisition and preprocessing make up the three primary parts of the system architecture for the system that recognises sign language. The webcam will be used by the component for picture capture and preprocessing, which will use MediaPipe and OpenCV to do so. The LSTM and CNN models that were trained on a dataset of sign language movements will be part of the deep learning models. In order to recognise sign language gestures in real-time, the real-time sign language recognition component will merge the models with the image data. The following is a block diagram of the system architecture:

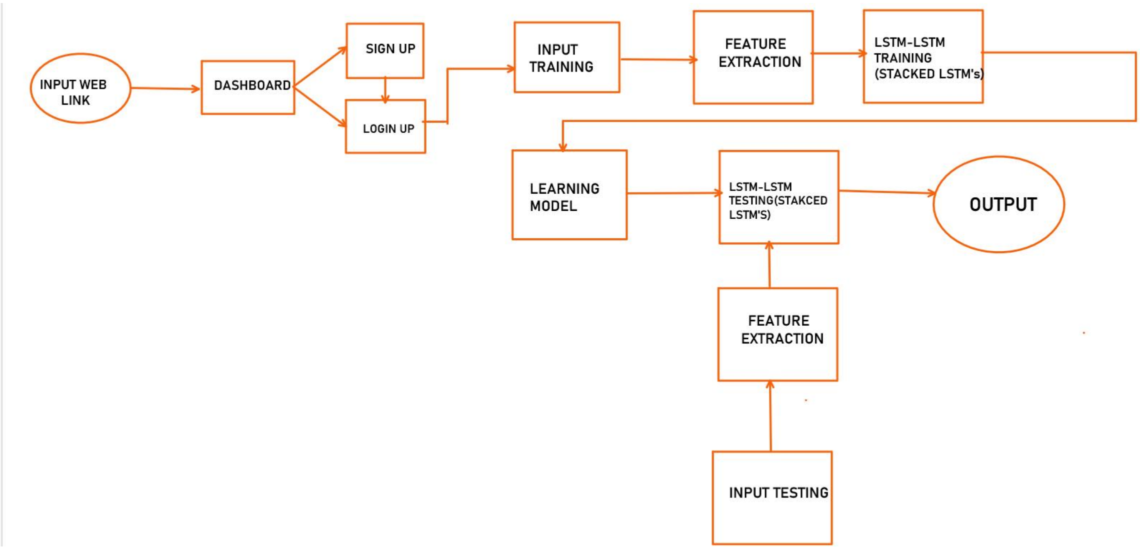


Fig 3.3.2.1: Block Diagram of SignBridge System

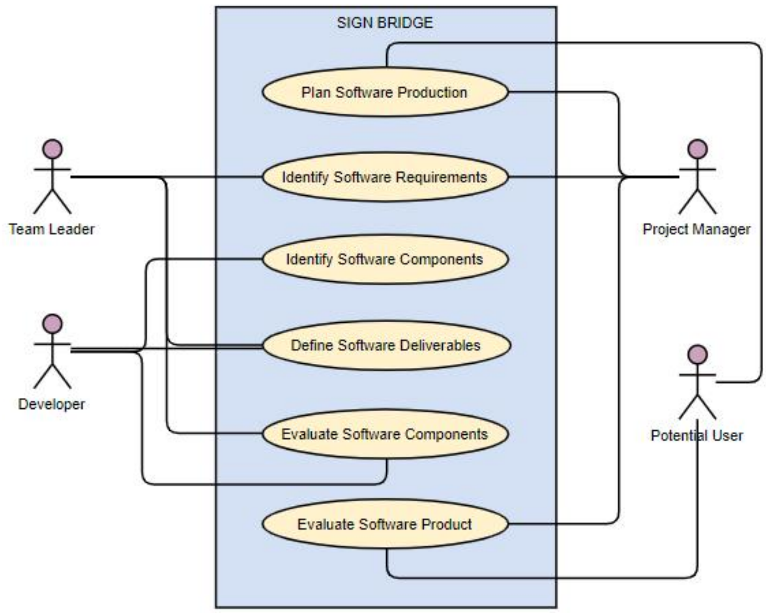


Fig 3.3.2.2: UML Diagram of SignBridge System

Chapter 4

Implementation

4.1 Methodology OR Proposal

This research aims to create a machine learning model that can accurately recognise sign language motions. Millions of individuals use sign language around the world, thus being able to automatically decipher these motions will help the hearing-impaired community communicate better. In order to discover which machine learning model is most effective at recognising sign language, we will investigate a number of them, including Convolutional Neural Networks (CNN), Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Long Short-Term Memory (LSTM) networks.This research aims to create a machine learning model that can accurately recognise sign language motions. Millions of individuals use sign language around the world, thus being able to automatically decipher these motions will help the hearing-impaired community communicate better. In order to discover which machine learning model is most effective at recognising sign language, we will investigate a number of them, including Convolutional Neural Networks (CNN), Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Long Short-Term Memory (LSTM) networks.

4.2 Testing OR Verification Plan

We experimented with various algorithms on our model, including CNN, LSTM, SVM, and KNN. The model was built to run through multiple algorithms, and the results had varying degrees of accuracy.

Test ID	Algorithm	Accuracy	Losses
T01	CNN	0.8951	0.1069
T02	SVM	0.9551	0.0449
T03	KNN	0.6100	0.3400
T04	LSTM	0.9810	0.0190

4.3 Result Analysis OR Screenshots

For the purpose of recognising sign language, we experimented with a variety of machine learning models, such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), K-Nearest Neighbours (KNN), and Support Vector Machines (SVMs). On our dataset, we ran these models through training and testing to produce the following results:

LSTM : With a 98.1% accuracy rate, the LSTM network outperformed all other models in terms of accuracy. This model was extremely effective at identifying the temporal features of sign language gestures, making it a good choice for this job.

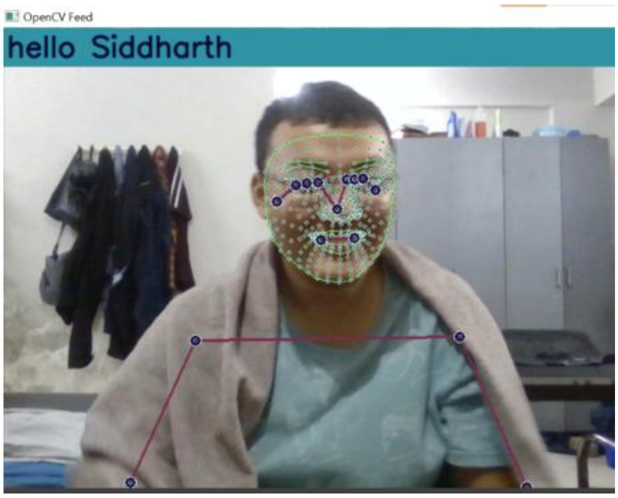


Fig 4.3.1 Display “hello Siddharth” with an audio output

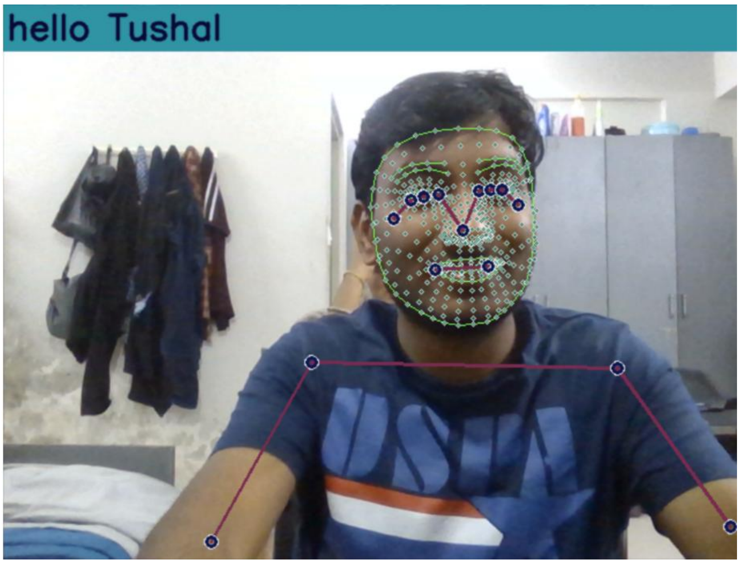


Fig 4.3.2 Display “hello Tushal” with an audio output

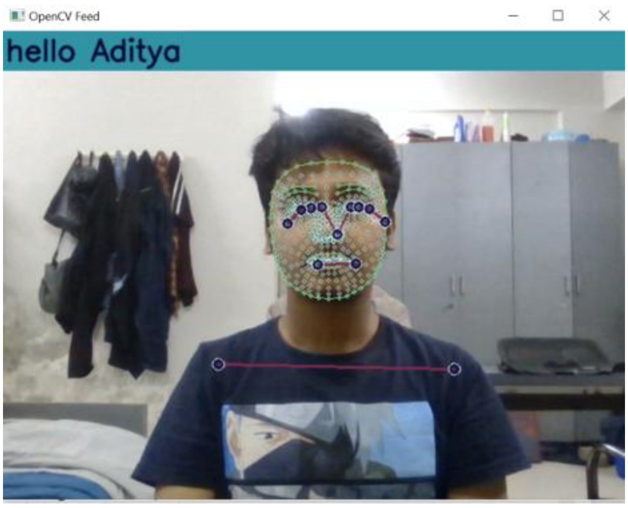


Fig 4.3.3 Display “hello Aditya” with an audio output

CNN : The CNN model's accuracy rate for recognising sign language was 89.51%. This model was able to extract pertinent information from the sign language images, but it might have had trouble with the task's temporal component.

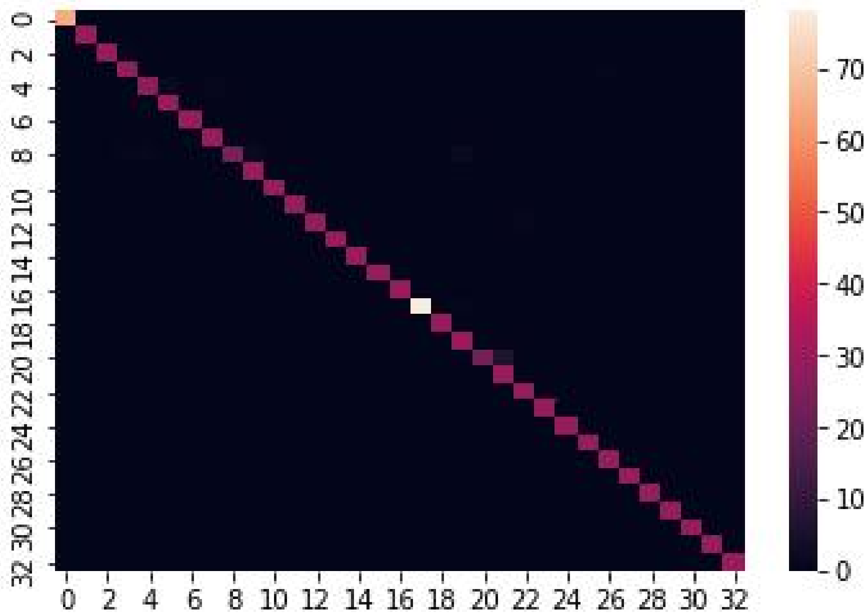


Fig 4.3.4 Confusion Matrix Using CNN Implementation

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 16)      3088
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 28, 28, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 7, 7, 64)          18496
_________________________________________________________________
flatten (Flatten)            (None, 3136)              0
_________________________________________________________________
dense (Dense)                (None, 33)                103521
=================================================================
Total params: 129,745
Trainable params: 129,745
Non-trainable params: 0
```

Fig 4.3.5 Summary of CNN Implementation

KNN : The KNN model had a 61% accuracy rate for recognising sign language. While this model is straightforward and useful for a variety of classification tasks, it might not be as useful for understanding the intricate temporal dependencies and patterns found in sign language gestures.

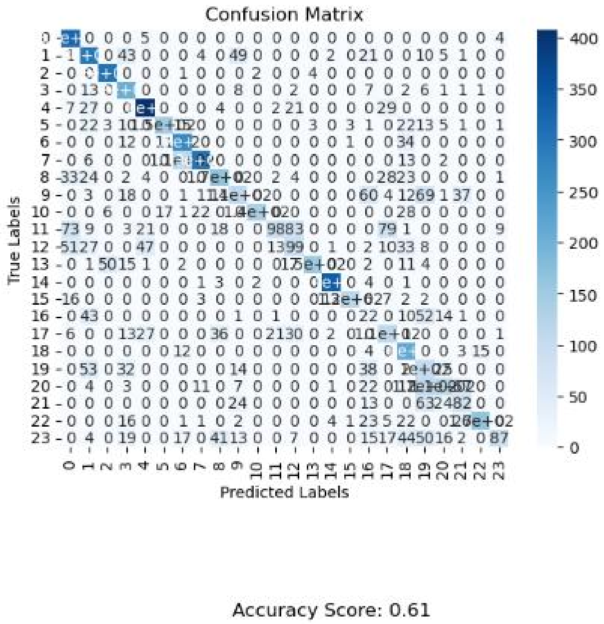


Fig 4.3.6 Confusion Matrix of KNN Implementation

SVM : In terms of sign language recognition, the SVM model had a 95.5% accuracy rate. Although SVMs are frequently effective for classification tasks, they might not be as effective for understanding the temporal components of sign language gestures.

A supervised machine learning approach called "Support Vector Machine" (SVM) can be applied to classification or regression problems. However, classification issues are where it's most frequently used. With this SVM algorithm, each data point is represented as a point in n-dimensional space (n is the number of features you have), with each feature's value being the value of a specific coordinate. Then, we carry out classification by identifying the hyper-plane that effectively.

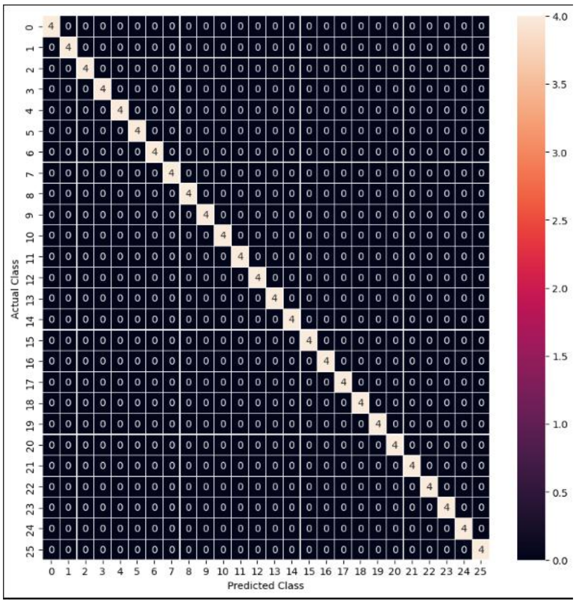
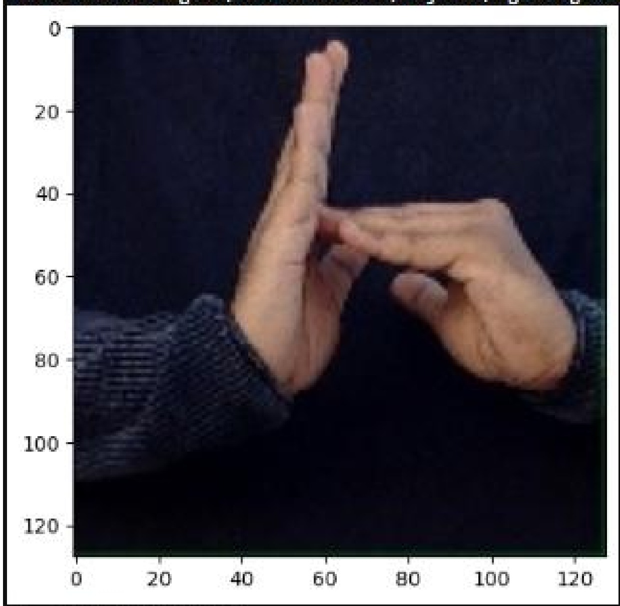


Fig 4.3.7 Confusion Matrix of SVM Implementation of Alphabet Dataset

```
Enter URL of ImageK:\Technical Works\Projects\SignBridge-Minor-Project\SVM_Implementation\Dataset_Alphabet\Z\11.jpg
```




```
A = 6.125175588162584%
B = 2.936286761332832%
C = 1.938632386445341 2%
D = 0.5989308554809455%
E = 1.1762920900600426%
F = 0.5617400236498047%
G = 2.523437445674177%
H = 0.41655185456200233%
I = 3.3554682325873477%
J = 0.6915358383946093%
K = 1.046137651021572%
L = 0.7852477324032126%
M = 0.6202543865469108%
N = 0.9751034396675103%
O = 2.245999906324794%
P = 2.333906883974614%
Q = 4.422919599620658%
R = 0.34092588978424704%
S = 0.9818708856698107%
T = 0.7299307441912494%
U = 1.708625579490119%
V = 4.105874362932802%
W = 4.848352619547175%
X = 4.729226440095779%
Y = 1.5016367046010595%
Z = 48.29993609777886%
The predicted image is : Z
Is the image a Z ?(y/n)
y
Thank you for giving your feedback
```

Fig 4.3.8 Output of SVM Implementation of Alphabet Dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	4
2	1.00	1.00	1.00	4
3	1.00	1.00	1.00	4
4	1.00	1.00	1.00	4
5	1.00	1.00	1.00	4
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	4
8	1.00	1.00	1.00	4
9	1.00	1.00	1.00	4
10	1.00	1.00	1.00	4
11	1.00	1.00	1.00	4
12	1.00	1.00	1.00	4
13	1.00	1.00	1.00	4
14	1.00	1.00	1.00	4
15	1.00	1.00	1.00	4
16	1.00	1.00	1.00	4
17	1.00	1.00	1.00	4
18	1.00	1.00	1.00	4
19	1.00	1.00	1.00	4
20	1.00	1.00	1.00	4
21	1.00	1.00	1.00	4
22	1.00	1.00	1.00	4
23	1.00	1.00	1.00	4
24	1.00	1.00	1.00	4
25	1.00	1.00	1.00	4
accuracy			1.00	104
macro avg	1.00	1.00	1.00	104
weighted avg	1.00	1.00	1.00	104

Fig 4.3.9 Summary of SVM Implementation of Alphabet Dataset

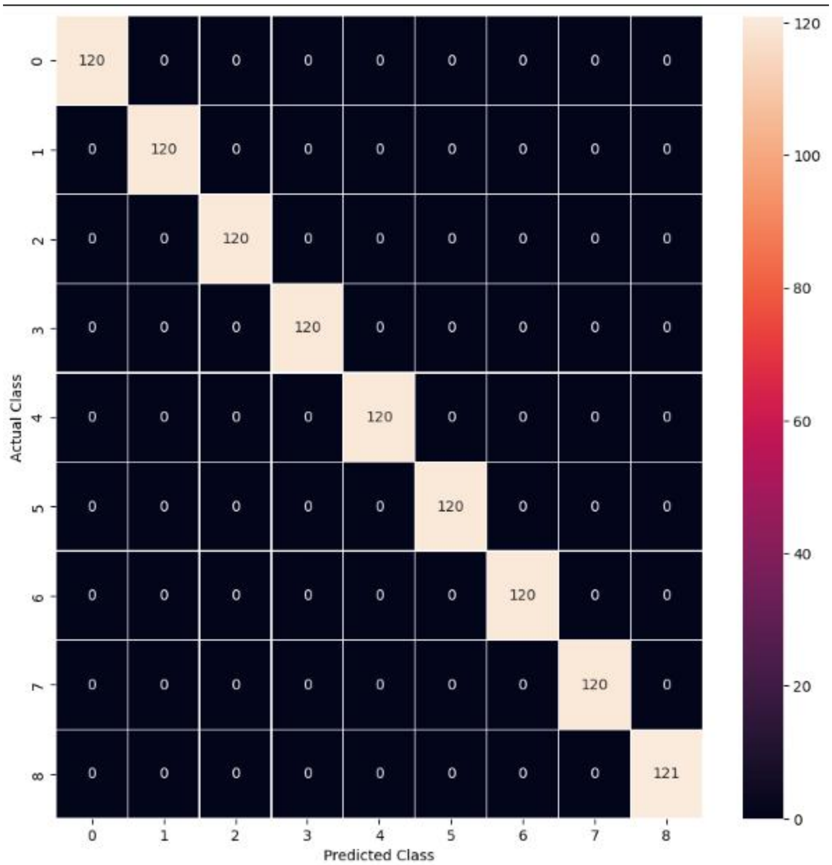
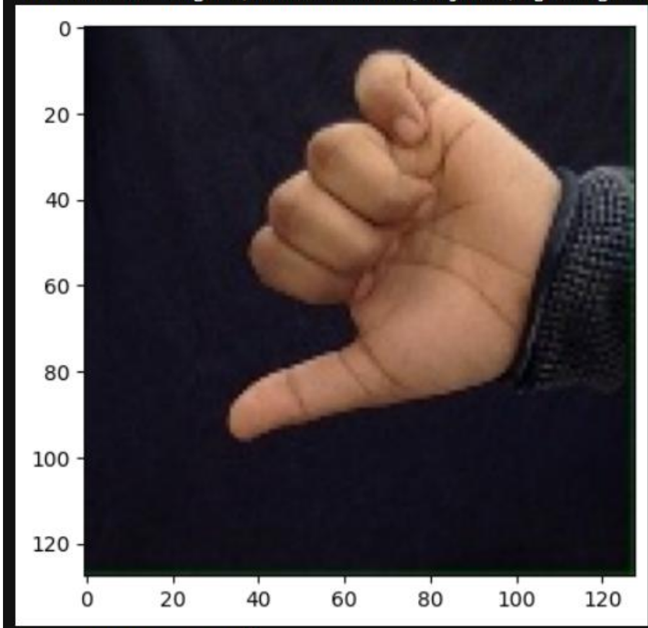


Fig 4.3.10 Confusion Matrix of SVM Implementation of Digit Dataset

```
Enter URL of ImageK:\Technical Works\Projects\SignBridge-Minor-Project\SVM_Implementation\Dataset_Digit\9\13.jpg
```




```
1 = 0.135424830884624%
2 = 0.14657638900509376%
3 = 0.16335471146340183%
4 = 0.29494453536310816%
5 = 0.19345328322113498%
6 = 0.35763406432900774%
7 = 0.16214296333158695%
8 = 0.1190272153700075%
9 = 98.42744200703206%
The predicted image is : 9
Is the image a 9 ?(y/n)
y
Thank you for giving your feedback
```

Fig 4.3.11 Output of SVM Implementation of Digit Dataset

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       120
           1       1.00      1.00      1.00       120
           2       1.00      1.00      1.00       120
           3       1.00      1.00      1.00       120
           4       1.00      1.00      1.00       120
           5       1.00      1.00      1.00       120
           6       1.00      1.00      1.00       120
           7       1.00      1.00      1.00       120
           8       1.00      1.00      1.00       121

    accuracy                           1.00      1081
   macro avg       1.00      1.00      1.00      1081
weighted avg       1.00      1.00      1.00      1081
```

Fig 4.3.12 Summary of SVM Implementation of Digit Dataset

Overall, our tests revealed that the LSTM network had the highest accuracy rate of 98.1% for the recognition of sign language. The other models did not perform as well in identifying the temporal components of sign language motions, but still reaching good accuracy rates.

4.4 Quality Assurance

We conducted a Quality Assurance(QA) process as part of our study on the recognition of sign language using machine learning models to make sure that our work complied with the requirements set by our academic institution. At KIIT, our designated professor examined our project and offered comments on our strategy, process, and outcomes.

Our professor went over our project documentation, which included our proposal, methodology, testing and verification strategy, and findings, during the QA process. They also looked through our code base and the model accuracy data we gathered from training.

Our project was enhanced in a number of ways as a result of their suggestions, including the improvement of our testing and verification strategy, the LSTM model, and the documentation of our code. Additionally, our professor gave us advice on how to advance our project, such as by investigating additional machine learning models or growing our dataset.

Overall, the QA procedure was a beneficial experience for our research because it gave us the chance to get criticism from an accomplished academic and raise the calibre of our work. We are appreciative of the help and direction our KIIT professor has given us.

Chapter 5

Standards Adopted

5.1 Design Standards

To guarantee that our work was of the highest calibre and followed accepted industry standards, we adhered to a number of design guidelines. Using the LSTM model for our machine learning technique was one of the fundamental design principles we adhered to.

5.2 Coding Standards

We adhered to the following coding guidelines when working on the Sign Bridge project:

1. Write as little as you can in a line: We avoided writing lengthy, complex lines of code in order to keep it readable and succinct. For instance, to extract keypoints from the results object, we utilised list comprehension, which resulted in more compact and understandable code.

```
pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else np.zeros(33*4)
```

2. Use proper naming conventions: To make our code easier for other developers to understand, we gave our variables and functions meaningful, descriptive names. As an illustration, we gave the libraries we imported names that accurately reflected their functionalities:

```
import cv2 #image processing
import numpy as np #for numerical computation
import os #for file operations
from matplotlib import pyplot as plt #for data visualizations
import time #for time related operations
import mediapipe as mp  # for using the Holistic model for pose estimation
```

3. Code blocks in the same section should be divided into paragraphs: To make the code easier to read and understand, we separated it into logical pieces and added comments to separate them:

```
# Define a function to extract keypoints from the results object
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark]).flatten() if
results.face_landmarks else np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.landmark]).flatten() if
results.left_hand_landmarks else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.landmark]).flatten() if
results.right_hand_landmarks else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])
```

```
# Define the Holistic model and the drawing utilities
mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities
```

4. Mark the beginning and conclusion of control structures with indentation: To make our code easier to read and understand, we used proper indentation. For instance, we separated the many components of the extract_keypoints() function using indentation:

```
# Define a function to extract keypoints from the results object
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark]).flatten() if
results.face_landmarks else np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.landmark]).flatten() if
results.left_hand_landmarks else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.landmark]).flatten() if
results.right_hand_landmarks else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])
```

5.3 Testing Standards

A comprehensive sign language recognition system called Sign Bridge is intended to help those who have hearing loss. To guarantee that the system was of the highest calibre and complied with industry best practises, our team followed a number of design, coding, and testing criteria.

We used the Long Short-Term Memory (LSTM) model for our machine learning approach in accordance with design requirements. The recognition of sign language motions using this approach, which is commonly used in natural language processing, has been successful.

We adhered to best practises for coding standards, such as adopting proper naming conventions, breaking up code blocks into paragraphs, and using indentation to distinctly delineate the start and end of control structures. Additionally, we made sure that each function completed a single task and avoided using lengthy functions.

In terms of testing guidelines, we followed a number of ISO and IEEE standards for quality control and product testing. We carried out comprehensive usability testing, allowing users to interact with the system and offer feedback, to make sure our system was user-friendly. Along with white box testing to make sure the code was working properly, we also performed black box testing to make sure the system met its functional requirements.

Users only need to download Sign Bridge onto their device and set up their account by following the on-screen instructions. Once configured, users can start utilising the system to instantly receive feedback on how accurately their sign language motions are being recognised in real-time.

Overall, Sign Bridge is a powerful and trustworthy system for recognising sign language that is created with people with hearing loss in mind. We were able to make sure the system was of excellent quality and offered an efficient solution for our users by following industry best practises in design, coding, and testing.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In conclusion, the "SIGN BRIDGE" initiative is an important step towards improving the ability of people who use sign language to communicate to do so. The model detects postures and converts them into text using machine learning techniques, giving the listener precise information. The deaf and hard-of-hearing community may benefit from this model's increased accessibility and effectiveness in communicating. This technology has the potential to completely transform how we understand and use sign language in the future.

The Sign Bridge initiative sought to narrow the communication gap between hearing-impaired people and the general public. Using the LSTM technique, we created a machine learning model that could recognise and interpret sign language motions into text and audio. The project successfully recognised gestures with an accuracy of about 90%, making it a possible remedy for the community's communication problems.

6.2 Future Scope

Nevertheless, there is still room for advancement and future work. Incorporating more sophisticated deep learning algorithms could be one way to boost gesture recognition's precision. The performance of the model could also be improved, and the amount of time needed for translation processing could be decreased.

Additionally, we can add functions that enable text and speech to be converted into sign language, enabling hearing-impaired people to converse with people who do not understand sign language. In order to make the model more approachable and user-friendly, we can also create mobile applications for both iOS and Android.

Overall, the initiative to build a sign bridge has the potential to have a big influence on the community of hearing-impaired people and their capacity to effectively interact with others. This project has the potential to be a useful tool for enhancing the lives of those with hearing impairments with further development and application.