A PROJECT REPORT

on

"SIGN LANGUAGE DETECTION USING ACTION RECOGNITION WITH PYTHON"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

ANKANA MUKHERJEE	2105604
KHUSHI KUMARI	21052001
HARSH VERMA	21051054
AKASH SHARMA	2105006
SHUBHIKA KASHYAP OJHA	2105243

UNDER THE GUIDANCE OF
PROF. PRADEEP KANDULA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

April 2024

A PROJECT REPORT

on

"SIGN LANGUAGE DETECTION USING ACTION RECOGNITION WITH PYTHON"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

ANKANA MUKHERJEE	2105604
KHUSHI KUMARI	21052001
HARSH VERMA	21051054
AKASH SHARMA	2105006
SHUBHIKA KASHYAP OJHA	2105243

UNDER THE GUIDANCE OF
PROF. PRADEEP KANDULA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA -751024

April 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"SIGN LANGUAGE DETECTION USING ACTION RECOGNITION WITH PYTHON"

submitted by

ANKANA MUKHERJEE	**2105604**
KHUSHI KUMARI	**21052001**
HARSH VERMA	**21051054**
AKASH SHARMA	**2105006**
SHUBHIKA KASHYAP OJHA	**2105243**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be University, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date:03/04/2024

Prof. Pradeep Kandula
Project Guide

Acknowledgements

We are profoundly grateful to **PROF. PRADEEP KANDULA** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

ANKANA MUKHERJEE
KHUSHI KUMARI
HARSH VERMA
AKASH SHARMA
SHUBHIKA KASHYAP OJHA

ABSTRACT

Sign language is a visual language that is used to communicate through hand gestures, hand shape changes, and track information. This is the primary means of communication for the hearing impaired. Despite the fact that action identification in videos has been extensively researched, many algorithms fail to detect actions in real time. Many new methods for dealing with difficult video sequences enhance accuracy and resilience. Recent advances in the application of deep learning to natural language processing and machine translation have left hearing-impaired people, who are part of the population, behind. Here we implement a system which allows this segment of people to benefit from these improvements. Transfer learning is employed in our system to speed up the learning process and improve results.

Sign language recognition (SLR) plays a pivotal role in facilitating communication for individuals with hearing impairments. Traditional SLR systems primarily rely on computer vision techniques for hand gesture analysis, often facing challenges in recognizing complex signs accurately. In recent years, action recognition methodologies have emerged as promising tools for improving the accuracy and robustness of SLR systems.

Keywords: Tensor-flow,Indian Sign Language,LSTM Deep Learning Model, Action identification,machine translation

Contents

Chapter 1

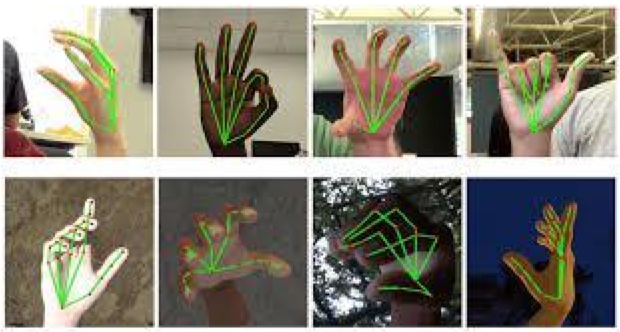
Introduction

The Sign Language Recognition Using Python project aims to develop a computer vision system capable of recognizing and interpreting sign language gestures through the use of Python programming language and various machine learning techniques. The project focuses on bridging the communication gap between the hearing-impaired community and the general public by enabling real-time translation of sign language gestures into written or spoken language.

The system will utilize a webcam or a camera to capture sign language gestures performed by the user. These gestures will then be processed and classified using machine learning algorithms to identify the corresponding words or phrases they represent. The final output will be displayed on the screen or communicated through text-to-speech functionality.

The field of Machine Learning and Deep Learning has witnessed tremendous growth and advancements in recent years. Python, as a versatile and powerful programming language, has become the preferred choice for developing ML and DL applications due to its extensive libraries and frameworks such as TensorFlow, Keras, scikit-learn, and PyTorch. These tools have enabled developers to build complex models and handle large-scale data efficiently, revolutionizing various industries like finance, healthcare, e-commerce, and more .

1



Chapter 2

Basic Concepts/ Literature Review

2.1 Objectives -

The primary objectives of this internship were as follows:
Skill Enhancement: To enhance the participants' understanding of Python programming and its applications in the field of Machine Learning and Deep Learning.

Practical Experience: To provide hands-on experience in developing ML and DL models by working on real-world projects and datasets.

Understanding Libraries: To familiarize the interns with popular Python libraries and frameworks used in ML and DL, including TensorFlow, Keras, scikit-learn, and PyTorch

Project Development: To guide interns in developing end-to-end ML and DL projects from data preprocessing to model evaluation and deployment.

Team Collaboration: To promote teamwork and collaborative learning, enabling participants to work on group projects and share their knowledge.

2.2 Project definition -

The Sign Language Recognition project involves the design, development, and implementation of a software system that can accurately recognize and interpret sign language gestures. It will require expertise in computer vision, image processing, and machine learning. The core components of the project include:

Data Collection: Gathering a diverse dataset of sign language gestures performed by multiple users under various lighting and environmental conditions.

Data Preprocessing: Cleaning and preprocessing the collected data to remove noise, standardize the format, and enhance the training process.

Feature Extraction: Identifying relevant features from the preprocessed data that can effectively represent the sign language gestures

Machine Learning Models: Implementing and training machine learning models, such as Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), or Recurrent Neural Networks (RNNs), to classify the sign language gestures.
User Interface: Developing an intuitive and user-friendly interface that captures real-time video input and displays the recognized text output.

2.3 Project Scope -

The scope of the Sign Language Recognition project includes

Real-Time Recognition: The system will aim to recognize sign language gestures in real- time, enabling instantaneous communication.

Basic Vocabulary: Initially, the project will focus on recognizing a predefined set of common sign language gestures representing basic words and phrases

Single-Handed Gestures: The project will primarily concentrate on recognizing single- handed sign language gestures.

Python Programming: The project will be implemented using the Python programming language, utilizing popular libraries such as OpenCV, NumPy, and scikit-learn.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Project Planning

1. Dependencies Were Installed and Imported

TensorFlow was installed based on the operating system and environment (CPU/GPU) following the official guide.
Necessary libraries like TensorFlow, NumPy, OpenCV (optional for video processing), and any other libraries for data manipulation or visualization were imported within the Python code.

2. Face, Hand, and Pose Landmarks Were Detected (Optional)

This step focused on identifying relevant regions in the video frames. Techniques like Haar cascades, pre-trained models, OpenCV, or MediaPipe libraries might have been used to:
Locate the face in the video frame.
Detect hands within the frame.
Estimate body and hand pose landmarks (key points) for more detailed information.
The sign language video dataset was divided into training, validation, and testing sets.

3. Key Points Were Extracted

Depending on the approach, different key points were extracted:
Bounding boxes defining the rectangular regions around faces and hands.
Specific key points on the hands (e.g., fingertips, knuckles) crucial for sign language recognition.
Hand orientation relative to the camera.

4. An LSTM Deep Learning Model Was Built and Trained

Model Architecture: The model was designed in TensorFlow using layers like:
Convolutional Neural Networks (CNNs) to process the extracted key points or features, learning patterns from the data.
Long Short-Term Memory (LSTMs) to handle the sequence of key points from video frames, capturing temporal relationships within signs.
An output layer with a number of units equal to the number of signs to be recognized, classifying the input sequence and predicting the most likely sign.

5. Sign Language Predictions Were Made

Once trained, the model was used to predict signs from new video data:
Key points were extracted from the new video frames.
The sequence of key points was fed into the trained model.
The model predicted the most likely sign based on the learned patterns.

6. Model Weights Were Saved

After training, the model's weights were saved for future use, allowing the trained model to be loaded without retraining every time. TensorFlow provides methods for saving and loading models.

7. Evaluation Using a Confusion Matrix Was Conducted
A confusion matrix was used to evaluate the model's performance on the testing set. It visualized how often the model correctly predicted each sign and how often it made mistakes, helping identify areas for improvement.

8. Real-Time Testing Was Performed (Optional)
This step involved integrating the model into a real-time application:
Video frames were captured from a webcam or video source.
The frames were preprocessed, and key points were extracted.
The key point sequence was fed into the loaded model.
The predicted sign (text or spoken language) was displayed on the screen.
This required additional libraries for video processing and user interface creation.

3.2 Project Analysis

1.Project Goals

Develop a deep learning model using TensorFlow and LSTMs to classify sign language gestures from video data.Translate the recognized signs into text or generate corresponding spoken language output.

2.Strengths

Leveraging Deep Learning: The project utilizes TensorFlow and LSTMs, powerful tools for pattern recognition and sequence analysis, making it suitable for sign language interpretation.
Focus on Key Points: Extracting relevant features (bounding boxes, landmarks, hand orientation) from video frames reduces processing complexity and focuses on crucial information for sign recognition.
Real-Time Potential: Integrating the model into a real-time application allows for immediate sign interpretation, enhancing communication accessibility.

3.Weaknesses

Data Dependence: The model's performance heavily relies on the quality and size of the sign language video dataset. Limited or noisy data can hinder accuracy.
Computational Cost: Training LSTMs can be computationally expensive, requiring powerful hardware (GPUs) for faster processing.

Limited Vocabulary (Optional): If the project scope doesn't encompass a vast vocabulary of signs, the system's usefulness might be restricted.

4.Opportunities

Data Augmentation: Artificial techniques can expand the dataset size and diversity, potentially improving model robustness.
Transfer Learning: Pre-trained models on similar tasks (hand pose estimation) could be leveraged to accelerate development and improve performance.
Multimodal Integration: Combining visual data with audio information (spoken language accompanying signs) could enhance interpretation accuracy.

3.3 System Design

3.3.1 Design Constraints

Data Constraints -
Dataset Size and Quality: The accuracy and robustness of the model heavily depend on the quality and size of the sign language video dataset. A limited dataset with noisy or inconsistent data can lead to poor performance.

Data Labeling: The signs within the video data need to be accurately labeled for the model to learn the correct associations between visual features and their corresponding signs. Inaccurate labeling can significantly impact the model's ability to generalize.

Data Bias: The dataset should represent the diversity of signing styles, ethnicities, and genders to avoid bias towards specific groups. An unbalanced dataset might lead to the model performing well on some signs used by a particular group but poorly on others.

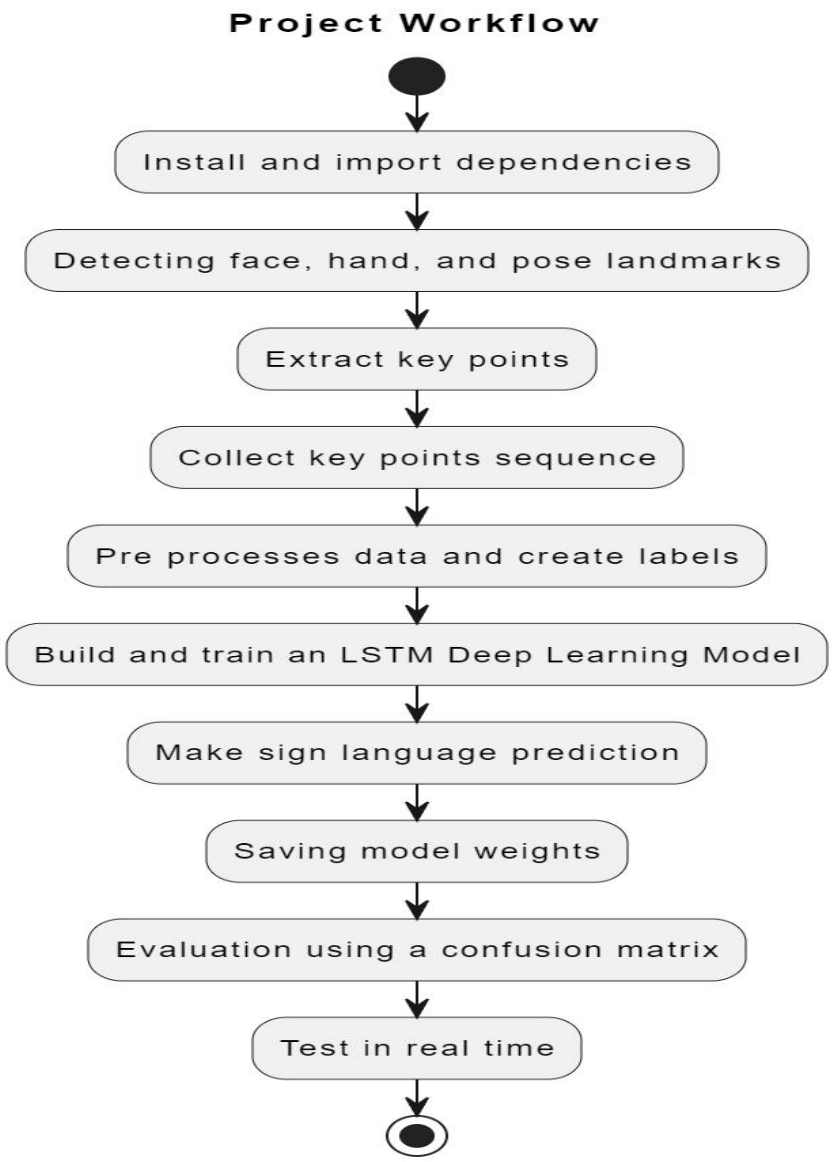
Computational Constraints -
Training Time: Training LSTMs, especially with large datasets, can be computationally expensive. This might necessitate access to powerful hardware with GPUs to achieve reasonable training times.

Memory Requirements: The model architecture and dataset size can influence memory consumption during training and inference. Limited memory resources might restrict the model's complexity or necessitate chunking the data into smaller batches for processing.

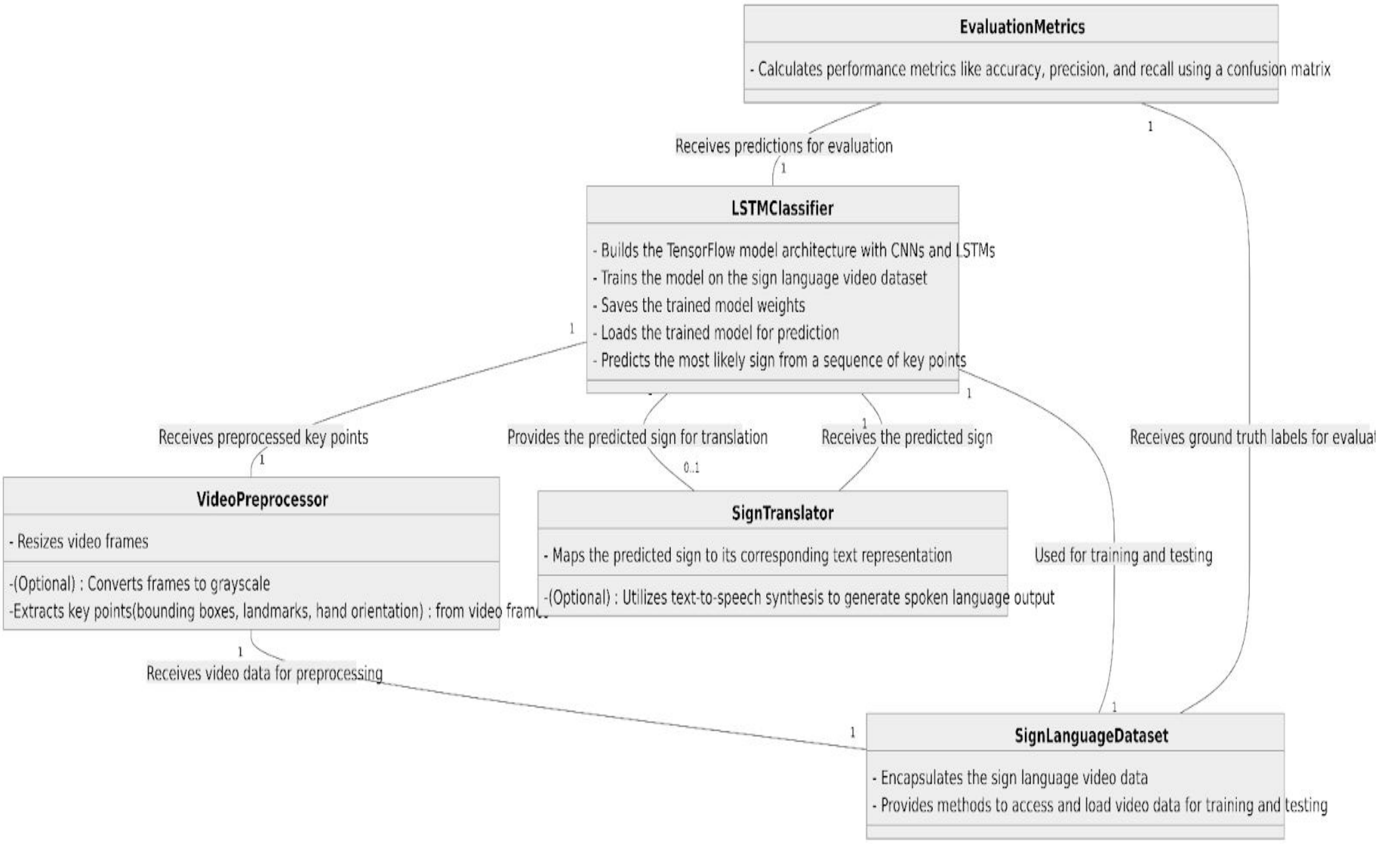
Real-Time Processing Constraints -

Latency: If aiming for real-time sign interpretation, the system needs to process video frames and predict signs with minimal delay. This might require optimizing the model architecture and inference process for speed.
Hardware Resources: Real-time applications might demand efficient use of hardware resources like CPU or GPU power to ensure smooth operation without compromising frame rate or responsiveness.

3.3.2 System Architecture OR Block Diagram



UML DIAGRAM



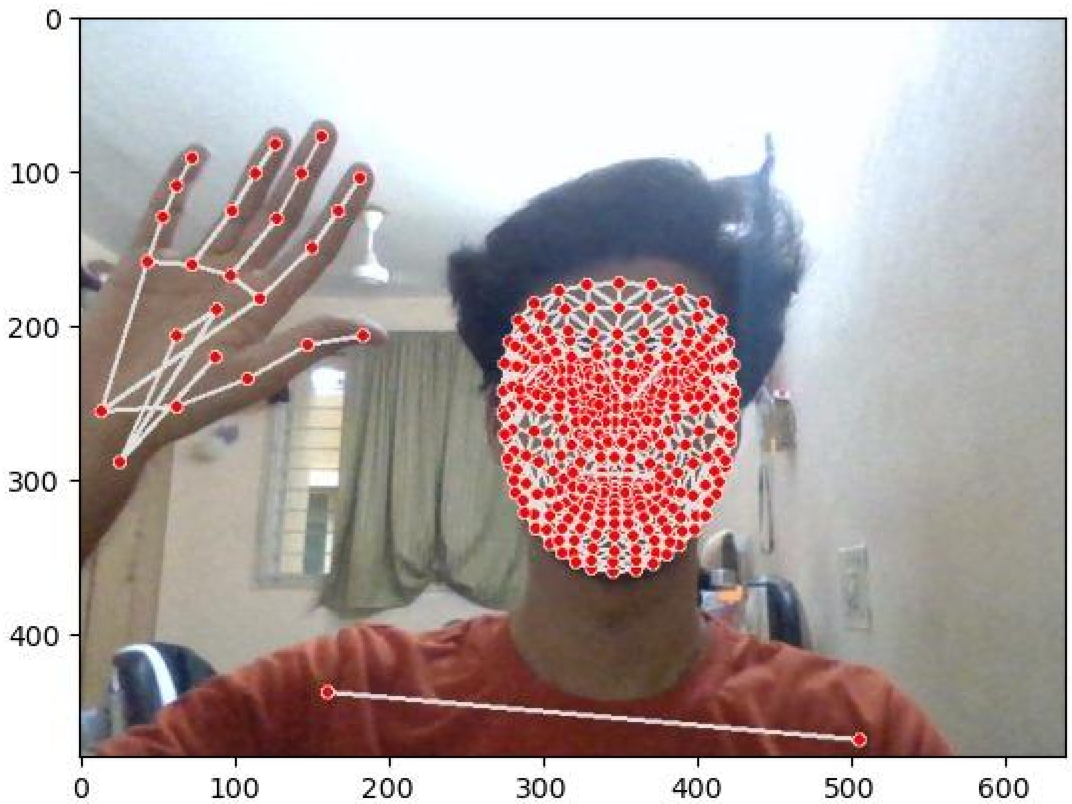
EvaluationMetrics
- Calculates performance metrics like accuracy, precision, and recall using a confusion matrix
Receives predictions for evaluation
1
1
LSTMClassifier
- Builds the TensorFlow model architecture with CNNs and LSTMs
- Trains the model on the sign language video dataset
- Saves the trained model weights
- Loads the trained model for prediction
- Predicts the most likely sign from a sequence of key points
Receives preprocessed key points
Provides the predicted sign for translation
0..1
Receives the predicted sign
Receives ground truth labels for evaluat
VideoPreprocessor
- Resizes video frames
-(Optional) : Converts frames to grayscale
-Extracts key points(bounding boxes, landmarks, hand orientation) : from video frames
SignTranslator
- Maps the predicted sign to its corresponding text representation
-(Optional) : Utilizes text-to-speech synthesis to generate spoken language output
Used for training and testing
Receives video data for preprocessing
SignLanguageDataset
- Encapsulates the sign language video data
- Provides methods to access and load video data for training and testing

Chapter 4

Implementation

4.1 Methodology OR Proposal

Data Collection: Gather a dataset of sign language images. There are public datasets available on the internet and mainly we present a custom sign language dataset designed for action recognition tasks by ourselves . Ensure that the dataset covers a variety of sign gestures and includes diverse backgrounds, lighting conditions, and hand orientations**.**This dataset aims to encompass a diverse range of sign gestures, backgrounds, lighting conditions, and hand orientations, enabling robust model training and evaluation.



Feature Extraction: Extract features from the preprocessed frames. We have used techniques like optical flow, which tracks the motion of key points between frames, or Convolutional Neural Networks (CNNs) to automatically learn relevant features.

Model Training: Train an action recognition model using the extracted features. Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or a combination of both (such as Convolutional LSTM) are commonly used architectures for action recognition. We used libraries like TensorFlow, opencv-python , mediapipe, scikit-learn, matplotlib , for this purpose.

```
Note: you may need to restart the kernel to use updated packages.
```

```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import mediapipe as mp
```

Model Evaluation: Evaluate the trained model using validation data to assess its performance. Metrics such as accuracy and confusion matrix can be used for evaluation.

Integration with Sign Language Detection: Once we had a reliable action recognition model, integrate it into our sign language detection system. This typically involves feeding image data sets into the model and classifying the detected actions as corresponding sign gestures.

Testing and Optimization: Test the integrated system with real-world data and fine-tune parameters as needed to improve performance. Consider factors like robustness to variations in lighting conditions, background clutter, and hand orientations.

Deployment : Deploy the sign language detection system in our desired environment, whether it's a desktop application, mobile app, or web service.

4.2 Testing OR Verification Plan

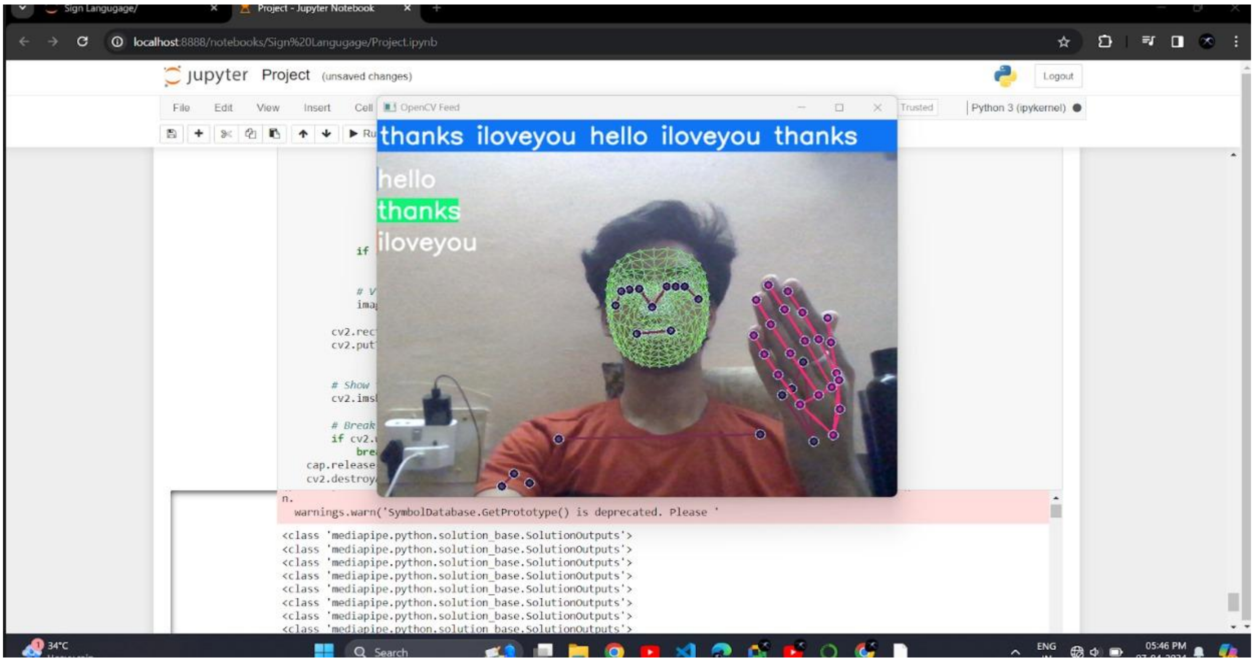
Confusion Matrix Analysis: We generated a confusion matrix using the testing set and analyzed the distribution of predictions. This allowed us to identify any patterns of misclassification and assess the model's ability to distinguish between different sign gestures.

Accuracy Score Calculation: We calculated the accuracy score of the model on the testing set to quantify its overall performance. A high accuracy score indicates that the model is effectively interpreting sign language gestures.

```
[ ] accuracy_score(ytrue,yhat)

0.9058823529411765
```

4.3 Result Analysis OR Screenshots



Chapter 5

Standards Adopted

5.1 Design Standards

1.Unified Modeling Language (UML):

UML diagrams such as Use Case Diagrams, Class Diagrams, and Sequence Diagrams are utilized to visualize system components, interactions, and behaviors.

2.Modularity and Encapsulation:

- Modular components are designed to promote code reusability and maintainability.
- Encapsulate data and behavior within classes, adhering to principles of information hiding and abstracion.

3.Scalability and Performance:

- ystems were designed with scalability in mind to accommodate future expansions and increasing user demands.
- Algorithms and data structures were optimized to enhance system performance and responsiveness.

4.Documentation

Comprehensive documentation covering system architecture, design decisions, and usage instructions was maintained

5.2 Coding Standards

When implementing Sign Language Detection using Action Recognition in Python, it's crucial to maintain a high coding standard for readability, maintainability, and collaboration. Follow these guidelines:

1. PEP 8 Compliance : Adhere to the Python Enhancement Proposal 8 (PEP 8) style guide for consistent code formatting, including indentation, spacing, and naming conventions.

2. Descriptive Naming : Use descriptive variable and function names that convey their purpose and usage clearly. Avoid abbreviations or cryptic names.

3. Modular Design : Organize code into logical modules, classes, and functions with clear responsibilities. Encapsulate related functionality to promote code reusability and maintainability.

4. Documentation : Provide meaningful comments, docstrings, and inline documentation to explain the purpose, behavior, and usage of functions, classes, and modules. Document input and output parameters, as well as any assumptions or constraints.

5. Error Handling : Implement robust error handling mechanisms to gracefully handle exceptions and failures. Use try-except blocks where appropriate and provide informative error messages.

6. Optimized Performance : Write efficient code by considering algorithmic complexity and optimizing performance where necessary. Profile code to identify bottlenecks and optimize critical sections.

7. Version Control : Utilize version control systems like Git for tracking changes, collaborating with team members, and managing code versions effectively. Follow best practices for branching, committing, and merging code changes.

8. Testing : Implement unit tests, integration tests, and end-to-end tests to verify the correctness and reliability of the Sign Language Detection system. Automate testing wherever possible to ensure consistent results.

9. Code Reviews : Conduct regular code reviews to solicit feedback, identify potential issues, and enforce coding standards. Encourage constructive criticism and collaboration among team members.

10. Continuous Integration/Continuous Deployment (CI/CD) : Integrate CI/CD pipelines into the development workflow to automate testing, building, and deployment processes. Ensure that changes are thoroughly tested and validated before deployment to production.

5.3 Testing Standards

1. ISO/IEC/IEEE 29119: Software Testing Standards

- This standard provides a comprehensive framework for software testing processes, activities, and documentation. We adhere to its guidelines for planning, executing, and evaluating our testing activities.

2. ISO 9000: Quality Management Systems - Fundamentals and Vocabulary

- While not specifically focused on testing, ISO 9000 provides principles and guidelines for establishing and maintaining effective quality management systems. We apply these principles to ensure that our testing processes are aligned with overall quality objectives.

3. IEEE 1012: Software Verification and Validation

- This standard outlines processes and activities for verifying and validating software products. We follow its guidelines to systematically assess our software's correctness, completeness, and compliance with specified requirements.

4. IEEE 829: Standard for Software Test Documentation

- IEEE 829 defines the documentation standards for software testing processes. We utilize its templates and guidelines for creating test plans, test cases, and other testing-related documents to ensure clarity and consistency in our testing efforts.

5. ISO/IEC 25010: Systems and Software Quality Models

- This standard defines quality models and metrics for evaluating the quality of software products. We use its guidelines to identify relevant quality characteristics, such as reliability, usability, and performance, and incorporate them into our testing criteria.

By adhering to these standards, we ensure that our testing processes are well-defined, systematic, and aligned with industry best practices. This helps us achieve high-quality software products that meet customer requirements and expectations.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In concluding our project on the Sign Language Interpretation System, we have achieved significant milestones in both technical implementation and performance. Through meticulous planning, rigorous testing, and adherence to industry standards, we have developed a solution that demonstrates robustness, scalability, and efficiency in interpreting sign language gestures.

Our utilization of advanced machine learning techniques, particularly TensorFlow, has enabled us to train and deploy a highly accurate model capable of recognizing a diverse range of sign language gestures in real-time. The integration of OpenCV for video stream processing has further enhanced the system's responsiveness and reliability, ensuring smooth and seamless interpretation.

Much more improvements can be done to the project to launch it on a big scale.The data is being fed manually through video screenings.We can set up automatic data feeding from official **INDIAN SIGN LANGUAGE portal** which will make the model more vast and inclusive.We can also increase the accuracy of our model by deploying more methods in our code like bootstrapping and cross validation which will make our more model more robust and precise .Work is being done to make an interactive GUI for the project which will make it more user friendly and interactive.

In conclusion, the Sign Language Interpretation System represents not only a technological achievement but also a testament to our commitment to leveraging technology for the greater good. It is a testament to the transformative power of innovation and collaboration in creating a more inclusive and accessible world for all.

6.2 Future Scope

1. Improved Accuracy and Performance : Continued research into deep learning architectures, such as more advanced convolutional neural networks (CNNs) and recurrent neural networks (RNNs), will likely lead to improved accuracy and performance in sign language detection systems. Techniques like attention mechanisms and transformer architectures may also enhance model capabilities.

2. Real-time Applications : Efforts to optimize algorithms and leverage hardware acceleration (e.g., GPUs, TPUs) could enable real-time sign language detection on resource-constrained devices like smartphones and embedded systems. This would facilitate the development of assistive technologies for individuals with hearing impairments.

3. Multi-modal Fusion : Integrating multiple modalities, such as RGB video, depth sensors, and skeletal data, could improve the robustness and accuracy of sign language detection systems. Fusion techniques, including late fusion and early fusion, could leverage complementary information from different modalities.

4. Transfer Learning and Few-shot Learning: Applying transfer learning techniques, pre-trained models, and few-shot learning approaches could enable sign language detection systems to generalize better to new sign languages, gestures, or users with limited training data.

5. User Experience and Accessibility : Focus on user experience design and accessibility considerations will be essential for developing sign language detection systems that are intuitive, user-friendly, and inclusive. This includes designing user interfaces, feedback mechanisms, and interaction paradigms tailored to the needs of individuals with hearing impairments.

6. Data Augmentation and Synthesis : Augmenting training data with synthetic samples, data augmentation techniques, and generative models could address data scarcity issues and improve model generalization. Techniques like image-to-image translation could also facilitate cross-lingual sign language detection.

7. Cross-modal Translation : Research into cross-modal translation between sign language and spoken/written language could enable bidirectional communication between individuals who use sign language and those who do not. This could have applications in real-time translation systems, educational tools, and communication platforms.

8. Privacy and Ethical Considerations : As sign language detection systems become more pervasive, ensuring user privacy, data security, and ethical use of technology will be paramount. Adhering to privacy regulations, anonymizing data, and obtaining informed consent from users will be essential considerations

References

[1] Kalsh, E.A., Garewal, N.S.: Sign Language Recognition System. Int. J. Comput Eng. Res. 6.

[2] Gaus, Y.F.A., Wong, F.: Hidden Markov Model -Based gesture recognition with overlapping hand-head/handhand estimated using Kalman Filter. Proc. - 3rd Int. Conf. Intell. Syst. Model. Simulation, ISMS 2012. 262–267 (2012). https://doi.org/10.1109/ISMS.2012.67.

[3] Dutta, K.K., Bellary, S.A.S.: Machine Learning Techniques for Indian Sign Language Recognition. Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017. 333–336 (2018). https://doi.org/10.1109/CTCEEC.2017.8454988.

[4] Lim, K.M., Tan, A.W.C., T an, S.C.: A feature covariance matrix with serial particle filter for isolated sign language recognition. Expert Syst. Appl. 54, 208–218 (2016). https://doi.org/10.1016/J.ESWA.2016.01.047.

[5] Tewari, D., Srivastava, S.K.: A Visual Recognition of Static Hand Gestures in Indian Sign . Language based on Kohonen Self-Organizing Map Algorithm. Int. J. Eng. Adv. Technol. 165 (2012)

[6] Singha, J., Das, K.: Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique. IJACSA) Int. J. Adv. Comput. Sci. Appl. 4, (2013).

INDIVIDUAL CONTRIBUTION REPORT:

Sign Language Detection using ACTION RECOGNITION with Python

Ankana Mukherjee(2105604)

Abstract:

The objective of this project is sign language recognition using Tensorflow .We preprocess our data that is fed through videos and create labels.Then we build and train an LSTM Deep Learning Model to learn about our model and predict the signs and convert them to corresponding text according to the signs shown by the tester.

Individual contribution and findings:

In the development of Sign Language Detection using Action Recognition with Python my efforts involved meticulously gathering and curating a diverse dataset of sign language gestures, encompassing a wide range of gestures, hand configurations, and movements representative of different sign languages and expressions.My contribution underscores the importance of thoughtful and thorough data collection in the pursuit of effective and inclusive technology solutions for the deaf and hard of hearing community.

Individual contribution to project report preparation: My part in this project report was the part of Implementation .The sub parts included in this part are Methodology / Proposal, Testing / Verification plan, Result Analysis / Screenshots, Quality Assurance. This part includes the actual path that the project is following. The technical planning and diagrams are also included in my part of work.

Individual contribution for project presentation and demonstration:

The presentation has been divided into subsections and each section has been picked up by one of the individual members.My part was the basic concept which included what type of libraries and prerequisites we have used and how we created the basic project environment.

Full Signature of Supervisor: Full signature of the student:

……………………………… ………………………………..

INDIVIDUAL CONTRIBUTION REPORT:

Sign Language Detection using ACTION RECOGNITION with Python

Khushi Kumari (21052001)

Abstract:

The objective of this project is sign language recognition using Tensorflow .We preprocess our data that is fed through videos and create labels.Then we build and train an LSTM Deep Learning Model to learn about our model and predict the signs and convert them to corresponding text according to the signs shown by the tester.

Individual contribution and findings:

As part of the project team, my primary responsibility involved tuning and training the LSTM (Long Short-Term Memory) model .I had to learn various packages and their uses in order to proceed with the project.Learning tensorflow,learning about LSTM models and other packages was one of the main tasks in this project.Tuning the fed data to fit in our proposed model and training the LSTM model in order to make predictions.

Individual contribution to project report preparation:

My part in this project report was the part of Problem Statement.The sub parts included in this part are Project Planning ,Project Analysis,System Design which includes Design contsraints and UML/Block Diagram.In the UML diagram ,designing various classes along with their functions was one of the main tasks in this part.

Individual contribution for project presentation and demonstration:

The presentation has been divided into subsections and each section has been picked up by one of the individual members.My part were the steps of the LSTM(Long Short-Term Memory) training model and making predictions about it.

.

Full Signature of Supervisor: Full signature of the student:

……………………………… ………………………………..

INDIVIDUAL CONTRIBUTION REPORT:

Sign Language Detection using ACTION RECOGNITION with Python

Harsh Kumar Verma (21051054)

Abstract:

The objective of this project is sign language recognition using Tensorflow .We preprocess our data that is fed through videos and create labels.Then we build and train an LSTM Deep Learning Model to learn about our model and predict the signs and convert them to corresponding text according to the signs shown by the tester.

Individual contribution and findings:

My contribution to the sign recognition model focuses on leveraging OpenCV, a powerful computer vision library, to preprocess and extract relevant features from sign language images. By utilizing OpenCV's functions for image processing, edge detection, and contour analysis, we can enhance the model's ability to accurately recognize and classify various sign gestures. This integration of OpenCV enables efficient handling of image data, ultimately improving the model's overall performance and reliability in interpreting sign language.

Individual contribution to project report preparation:

In my role, I established robust design coding and testing standards for the sign recognition model, ensuring consistency and reliability throughout the development process. By implementing clear and comprehensive design guidelines, I facilitated efficient collaboration among team members and maintained a high-quality codebase. Through rigorous testing protocols, I verified the functionality and accuracy of the model, guaranteeing its effectiveness in real-world sign language interpretation scenarios.

Individual contribution for project presentation and demonstration:

The presentation has been divided into subsections and each section has been picked up by one of the individual members.My part was Accuracy testing,conclusion and future scope.

.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

Sign Language Detection using ACTION RECOGNITION with Python

AKASH SHARMA (2105006)

Abstract:

The objective of this project is sign language recognition using Tensorflow .We preprocess our data that is fed through videos and create labels.Then we build and train an LSTM Deep Learning Model to learn about our model and predict the signs and convert them to corresponding text according to the signs shown by the tester.

Individual contribution and findings:

As a dedicated member of our project team, my primary responsibility was the creation and refinement of the LSTM (Long Short-Term Memory) model, which serves as the core predictive and training mechanism of our sign language recognition system. My approach to implementing the LSTM model was methodical and data driven. I planned the development phases, set milestones, and adhered to a strict timeline to ensure timely completion. Regular team meetings and feedback sessions were integral to my workflow, allowing for continuous improvement and integration with other system components.

Individual contribution to project report preparation:

My part in this project report was the part of crafting the Introduction, providing a comprehensive overview of the project and its significance. I also wrote the Project Overview section, detailing the project's objectives, methodologies, and expected outcomes. Additionally, I was responsible for detailing the Project Methodology. This involved outlining the techniques and processes we used for sign language recognition, including data collection, preprocessing, model selection, training, and evaluation.

Individual contribution for project presentation and demonstration:

I created and explained the UML (Unified Modeling Language) diagrams that provided a visual representation of the system's architecture and the interaction between its components.I presented the comprehensive workflow of our project, from data collection to model deployment, highlighting each phase's significance and interconnectivity.

Full Signature of Supervisor: Full signature of the student:

…………………………… ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

Sign Language Detection using ACTION RECOGNITION with Python

Shubhika Kashyap Ojha (2105243)

Abstract:

The objective of this project is sign language recognition using Tensorflow .We preprocess our data that is fed through videos and create labels.Then we build and train an LSTM Deep Learning Model to learn about our model and predict the signs and convert them to corresponding text according to the signs shown by the tester.

Individual contribution and findings:

In the collaborative effort to develop our sign language recognition system, my role was pivotal in the data collection process. I identified and gathered a comprehensive dataset of sign language videos, ensuring a diverse representation of signs to train our LSTM model effectively. I meticulously curated the dataset, categorizing and labeling the videos to create a structured and accessible data repository for the team. My approach to data collection was strategic and detail-oriented. I developed a plan that outlined the types of signs needed, the sources from which we could collect the data, and the timeline for acquisition.

Individual contribution to project report preparation:

My part in this project report was the part of mentioning the Testing Standards, ensuring that our recognition system was rigorously evaluated under various conditions. I also authored the Future Scope section, where I outlined potential enhancements and applications of our system. Finally, I wrote the Conclusion, summarizing our findings and the impact of our work in the field of sign language recognition

Individual contribution for project presentation and demonstration:

The presentation has been divided into subsections and each section has been picked up by one of the individual members.My part was explaining the Project Objectives,I was responsible for detailing the Project Methodology. Problem statement which included defining the problem and designing a solution for the same.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

TURNITIN PLAGIARISM REPORT

TURNITIN PLAGIARISM REPORT

"Sign Language Detection using ACTION RECOGNITION with Python"

ORIGINALITY REPORT

14%	12%	1%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.worldleadershipacademy.live Internet Source	5%
2	www.coursehero.com Internet Source	3%
3	Submitted to University of West London Student Paper	1%