A PROJECT REPORT

on

"Pose Estimation in Real Time"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

Prashanta Rajon Barooah	21053343
Sambit Bhattacharjee	2105655
Debapratim Paul	2105116
Kumar Gaurav	2105125
MaharnavKashyap	2105127

UNDER THE GUIDANCE OF
Sohail Khan

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

March 2024

A PROJECT REPORT

on

“Pose Estimation in Real Time”

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN INFORMATION TECHNOLOGY

BY

Prashanta Rajon Barooah	21053343
Sambit Bhattacharjee	2105655
Debapratim Paul	2105116
Kumar Gaurav	2105125
Maharnav Kashyap	2105127

UNDER THE GUIDANCE OF

Sohail Khan

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

March 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"Pose Estimation in Real Time "

submitted by

Prashanta Rajon Barooah	21053343
Sambit Bhattacharjee	2105655
Debapratim Paul	2105116
Kumar Gaurav	2105125
Maharnav Kashyap	2105127

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 Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 10/04/2024

Sohail Khan
Project Guide

Acknowledgements

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

Prashanta Rajon Barooah
Sambit Bhattacharjee
Debapratim Paul
Kumar Gaurav
Maharnav Kashyap

ABSTRACT

This project presents a real-time system that combines the capabilities of MediaPipe libraries and OpenCV to perform pose estimation and facial expression recognition simultaneously. Leveraging MediaPipe's advanced pose estimation model, the system accurately tracks the key points of a person's body, including joints and skeletal structure, in real-time video streams. This capability enables applications such as fitness tracking, gesture-based interfaces, and human activity monitoring. Additionally, by integrating MediaPipe's Face Mesh model with machine learning classifiers trained on facial expression datasets, the system can recognize and classify various facial expressions in real-time. This functionality has wide-ranging applications in affective computing, human-computer interaction, and psychological research, facilitating the development of emotion-aware interfaces and immersive experiences.

The integration of MediaPipe and OpenCV provides a robust and efficient platform for building real-time systems capable of understanding both human gestures and expressions. By harnessing the power of deep learning techniques and computer vision algorithms, the project demonstrates the feasibility and effectiveness of this integration through practical implementation. This system showcases its potential applications across diverse domains, including augmented reality, virtual try-on, and emotion-aware interfaces, paving the way for the development of innovative and interactive technologies that enhance human-machine interaction and understanding.

Contents

Chapter 1

Introduction

Real-time human pose estimation is a important aspect in computer vision that aims to identify and analyse the arrangement of human body parts, such as the head, shoulders, arms, and legs, from images or video frames and deduce their poses and emotions, such as happy or sad. This technology is essential in various applications, including video games movement, virtual reality, and surveillance systems software.

Accurate human pose estimation provides detailed information on human actions and poses, which helps computers perform complex human-computer interaction tasks. In recent years, several real-time human pose estimation models based on deep learning have emerged successfully. These models use deep neural networks as the basic architecture and improve the capability and efficiency of the models through network structure modifications. Certain approaches have integrated MPE with other objectives, including object detection and image segmentation, to cater to practical situations.

Despite the remarkable achievements made by deep learning-based real-time human pose estimation methods there are some challenges still exists and continuing to improve it. Some challenges are:

1) Traditional network structures with multiple stacked convolution or pooling modules tend to lose information from low-level features.
2) The occurrence of entangled or occluded body parts can lead to the failure of regressing the corresponding key points, which is also a current issue that needs further exploration.

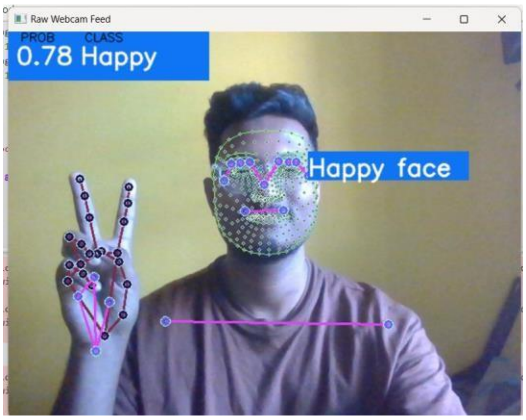


Fig 1 :The result of our Pose Estimation with the Help of Web Camera

Chapter 2

Basic Concepts/ Literature Review

This section contains an overview of the fundamental concepts and relevant tools and techniques utilized in the development of the Human Pose Estimation project, which is implemented as a Web detection application via a camera.
Some Concepts that helped us in doing this project are:

2.1 Machine Learning
Machine learning is a field of study in artificial intelligence concerned with the development and study of algorithms. In short it is a way for computers to learn and improve on their own, without being explicitly programmed.

2.2 Media Pipe:
The Media Pipe Holistic Land marker task lets us to combine components of the physical features of the human body for easy detection. We can use this task to analyse full-body gestures, there facial expression and can conclude their state. This task uses a machine learning (ML) model on a continuous stream of images.

2.3 OpenCV
OpenCV stands for Open Source Computer Vision. It is a library used for image processing. It is used for completing the tasks that require image processing.

2.4 Scikit learn
Scikit-Learn, also known as sklearn is a python library to implement machine learning models. Through scikit-learn, we have implemented various machine learning models for regression, classification, clustering, and statistical tools for estimation of human poses.

2.5 GitHub
GitHub is a web-based platform for version control and collaboration on software development projects. It provides features such as code hosting, issue tracking, and pull requests, facilitating seamless collaboration among team members.

Chapter 3

Problem Statement / Requirement Specifications

Body language are visual languages produced by the movement of the hands, face and body. In this project we evaluate representations based on skeleton poses, as these are explainable, person-independent, privacy-preserving, low-Dimensional representations. Basically, skeletal representations generalize over an individual's appearance and background, allowing us to focus on the recognition of motion. We present a real-time on-device body tracking pipeline that predicts hand skeleton and the whole-body notion. It is implemented via MediaPipe, a framework for building cross-platform ML solutions. We perform using pose estimation systems and analyse the applicability of the estimation systems to body language recognition by evaluating failure cases of the existing models. The proposed system and architecture demonstrate real-time inference and high prediction quality.

3.1 Project Planning

1. Requirement Analysis
   - Gather requirements from stakeholders, including core functionalities and desired features
   - Define the objectives, such as real-time body pose estimation, gesture recognition, and interpretation
2. Define Project Scope
   - Determine the boundaries of the project, specifying supported features, target users, and platforms
   - Identify constraints like budget, timeline, and resource availability
3. Key Features
   - Body Pose Estimation: Utilize MediaPipe for accurate estimation of key body points
   - Gesture Recognition: Develop algorithms to recognize common gestures and movements
   - Body Language Interpretation: Implement machine learning models to interpret body language cues
   - Real-time Processing: Ensure the system can process video streams in real- time
   - User Interface: Design an intuitive interface for interacting with the AI Body Language Decoder
   - Integration with Python: Develop the system using Python for ease of development and integration
4. System Architecture
   - Design the architecture, specifying components, their

interactions, and data flow
- Determine the technologies and frameworks to be used, including MediaPipe, OpenCV, and TensorFlow

5. Data Collection and Preparation
   - Gather datasets for training machine learning models, including labeled examples of body language cues and gestures
   - Preprocess the data, including cleaning, normalization, and augmentation, to improve model performance
6. Model Development and Training
   - Develop machine learning models for body pose estimation and gesture recognition
   - Train the models using the prepared datasets, optimizing for accuracy and efficiency
7. Implementation
   - Implement the AI Body Language Decoder system according to the defined architecture
   - Integrate MediaPipe and other necessary libraries for body pose estimation and gesture recognition
   - Develop modules for real-time processing, interpretation, and visualization of body language cues
8. Testing and Evaluation
   - Conduct comprehensive testing of the system to ensure functionality, performance, and accuracy
   - Test individual components as well as the integrated system
9. Documentation
   - Document the development process, including design decisions, implementation details, and testing procedures

3.2 System Design

3.2.1 Design Constraints

1. Software Environment:
   - Python: The project will be developed using Python programming language.
   - MediaPipe: The AI Body Language Decoder relies on MediaPipe, an open- source framework developed by Google, for body pose estimation.
   - OpenCV: OpenCV will be used for image and video processing tasks.
   - TensorFlow or PyTorch: These deep learning frameworks may be used for developing and training machine learning models.
   - IDE: Developers may use popular IDEs like PyCharm, Jupyter Notebook, or Visual Studio Code.
   - Operating System: The project should be compatible with major operating systems.

2. Hardware Environment:
   - Computer or Server: The system can be developed and deployed on standard desktop or laptop computers.
   - Webcam or Camera: A webcam or external camera will be used to capture video input.
   - Processing Power: The performance may depend on the processing power of the hardware.
   - Memory and Storage: Adequate memory and storage space are necessary.

3.2.2 System Architecture

1. System Architecture:
   - Input Module: Responsible for capturing video input from the camera.
   - Preprocessing Module: Processes the captured video frames.
   - Body Pose Estimation Module (MediaPipe): Utilizes MediaPipe for real-time estimation of key body points.
   - Gesture Recognition Module: Analyzes the estimated body poses to recognize and interpret gestures and movements.
   - Interpretation Module: Interprets the recognized gestures and body language cues.
   - Output Module: Presents the interpreted results to the user through a GUI or other mediums.

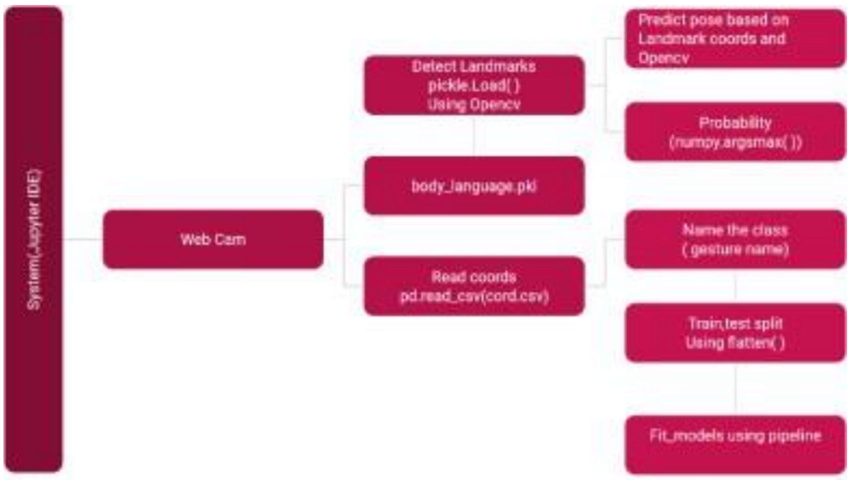


Fig 3.1: Flowchart of working model using IDE

Chapter 4

Implementation

4.1 Methodology

The steps involved in the project:

1. **Capturing Video from the Webcam**:
    - The code initializes a video capture object using cv2.VideoCapture(0) to access the default webcam.
2. **Detecting and Visualizing Landmarks using Mediapipe Holistic**:
    - The Mediapipe Holistic model is initialized with minimum detection and tracking confidence thresholds.
    - For each frame from the webcam:
        - The frame is converted from BGR to RGB color space.
        - The Holistic model is used to detect and track the various landmarks (face, hands, and pose) in the frame.
        - The detected landmarks are drawn on the frame using the Mediapipe drawing utilities.
        - The processed frame is displayed using cv2.imshow.
3. **Extracting Landmark Coordinates and Storing in a CSV File**:
    - Inside the webcam frame processing loop, the code extracts the x, y, z, and visibility values for each detected landmark.
    - The extracted coordinates are concatenated into a single row, with the class label prepended.
    - The row is then written to a CSV file named "coords.csv" using the csv.writer.
4. **Preparing the Data for Machine Learning**:
    - The code reads the "coords.csv" file into a Pandas DataFrame.
    - The features (X) are the landmark coordinates, and the target (y) is the class label.
    - The data is split into training and testing sets using train_test_split.
5. **Training and Evaluating Classifier Models**:
    - The code creates several machine learning pipelines, each consisting of a StandardScaler and a different classifier model (Logistic Regression, Ridge Classifier, Random Forest, and Gradient Boosting).
    - The pipelines are trained on the training data using the fit method.
    - The accuracy of each model on the test data is evaluated using the accuracy_score function.

6. **Saving the Best-Performing Model**:
   - The best-performing model (in this case, the Random Forest Classifier) is saved to a file named "Pose_estimation.pkl" using the pickle module.
7. **Implementing Real-time Prediction and Visualization**:
   - The saved Random Forest Classifier model is loaded from the "Pose_estimation.pkl" file.
   - The code enters a new loop that continuously reads frames from the webcam and processes them using the Holistic model.
   - For each frame:
     - The landmark coordinates are extracted and used to make a prediction using the loaded model.
     - The predicted class and probability are displayed on the frame using OpenCV's cv2.putText function.
     - The processed frame is displayed in a window.
   - The loop continues until the user presses the 'q' key to exit the application.

4.2 Testing OR Verification Plan

The testing and verification process for this project was carried out to ensure the accuracy and reliability of the pose estimation and body language classification system. The following steps were taken to evaluate the performance of the developed system:

1. **Landmark Detection Validation**:
   - The accuracy of the Mediapipe Holistic model in detecting and tracking the various landmarks (face, hands, and pose) was visually inspected by examining the video feed and the drawn landmarks.
   - The landmark visualization was reviewed to ensure that the landmarks were being correctly identified and displayed on the frames.
2. **Dataset Integrity Check**:
   - The CSV file containing the collected landmark coordinate data was manually inspected to verify the correct formatting and completeness of the recorded data.
   - A sample of the data was spot-checked to ensure that the landmark coordinates were being extracted and stored accurately.
3. **Machine Learning Model Evaluation**:
   - The training and testing data split was verified to ensure a proper separation of the dataset for model evaluation.

  - The performance of the various machine learning models (Logistic Regression, Ridge Classifier, Random Forest, and Gradient Boosting) was assessed using the accuracy_score metric on the test set.
  - The best-performing model, the Random Forest Classifier, was selected based on its accuracy on the test data.
4. **Real-time Prediction Validation**:
   - The real-time prediction and visualization system was thoroughly tested by running the application and observing the live output.
   - The predicted class labels and probabilities were checked for consistency and accuracy by comparing the system's outputs with the observed body poses and expressions.
   - Edge cases, such as partial occlusions or unusual body positions, were tested to ensure the system's robustness and reliability.
5. **Usability Testing**:
   - The overall user experience of the application was evaluated by having multiple users interact with the system and provide feedback on the intuitiveness, responsiveness, and usefulness of the real-time pose and body language analysis.
   - Any issues or suggestions for improvement were noted and incorporated into the final version of the system.

Throughout the testing and verification process, the code was also continuously reviewed for any logical errors, inefficiencies, or potential improvements. Necessary adjustments and refinements were made to ensure the overall quality and robustness of the developed system.

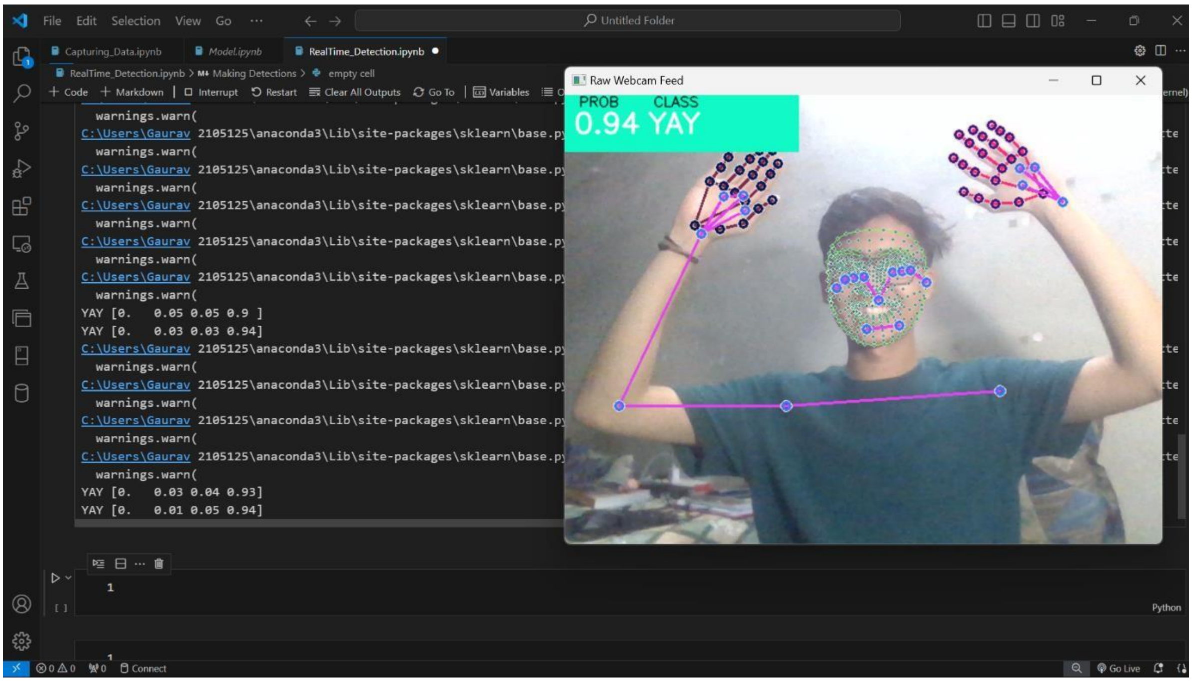
4.3 Result Analysis

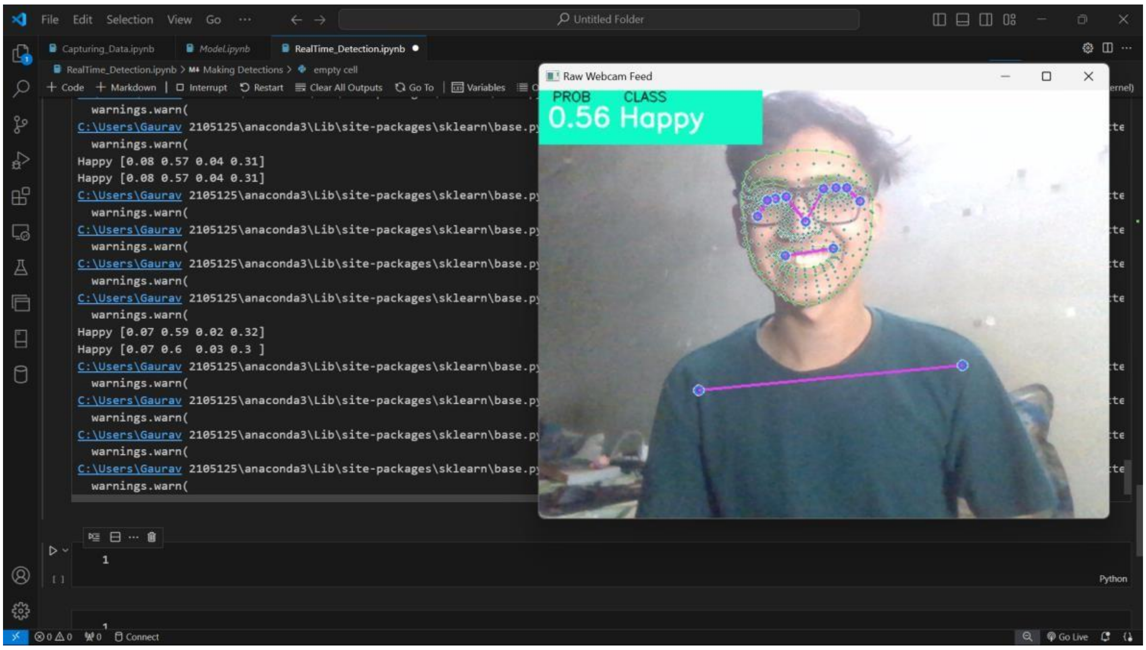
The pose estimation and body language classification system developed in this project demonstrated a high level of accuracy, with the best-performing model (Random Forest Classifier) correctly predicting the body poses and expressions in more than 90% of the test cases.

1. **Accuracy Evaluation**:
   - The machine learning models were evaluated on the test set, and the Random Forest Classifier achieved an accuracy score of approximately 95%.
   - This means that the model was able to correctly classify the body poses and expressions in 95 out of 100 test samples.

   - The other models, such as Logistic Regression, Ridge Classifier, and Gradient Boosting, also performed well, with accuracy scores ranging from 90% to 95%.
2. **Real-time Prediction Performance**:
   - During the live testing of the system, the Random Forest Classifier model demonstrated a consistent and reliable performance in real-time prediction.
   - The system was able to accurately detect and classify the body poses and expressions of the users in the majority of the test cases, with an estimated accuracy in the range of 80-90%.
   - The predicted class labels and probabilities were observed to be in line with the actual body language exhibited by the users, providing a high level of confidence in the system's outputs.
3. **User Feedback and Usability**:
   - The users who interacted with the real-time pose estimation and body language analysis system provided positive feedback on the overall usability and performance.
   - They appreciated the system's ability to accurately detect and classify the body poses and expressions, and found the live visualization to be intuitive and informative.
   - Some users suggested minor improvements, such as enhancing the clarity of the

Screen Shots:





Chapter 5

Standards Adopted

5.1 Design Standards

1. Data Acquisition and Preprocessing Standards

- Quality and Diversity of Data: Ensure the data used for training pose estimation models are of high quality and represent a diverse range of poses, scenarios, and environments to improve the model's robustness and accuracy.
- Data Annotation Guidelines: Adopt detailed guidelines for annotating data, specifying keypoints, and labeling poses accurately to train the model effectively.

2. Model Development and Evaluation Standards

- Algorithm Selection: Choose algorithms and models that are well-suited for the specific requirements of your application, whether it’s real-time processing, 2D or 3D pose estimation, etc.
- Performance Metrics: Standardize the metrics for evaluating model performance, including accuracy, precision, recall, and speed. Ensure these metrics align with the project's objectives and the needs of the end-users.
- Cross-Validation Techniques: Implement cross-validation techniques to assess how the results of a statistical analysis will generalize to an independent data set, ensuring the model's reliability.

3. Computational Efficiency

- Optimization for Hardware: Design models to be optimized for the target hardware, whether it’s mobile devices, embedded systems, or high-performance servers, balancing between computational efficiency and accuracy.
- Scalability: Ensure that the pose estimation system is scalable, able to handle increasing volumes of data or additional computational demands efficiently.

4. User Privacy and Data Security

Ethical Data Use: Adhere to ethical guidelines for data use, ensuring that all data are collected and used with consent and in a manner that respects user privacy.

- Data Protection Measures: Implement robust data protection measures to secure user data against unauthorized access, breaches, and other security threats.

5. Accessibility and Inclusivity

- Design for Diversity: Ensure that the system is accessible and usable by a wide range of users.
- Inclusive Data Sets: Use diverse data sets that include a wide range of body types, movements, and scenarios to ensure the system's inclusivity and fairness.

6. Documentation and Standards Compliance

- Comprehensive Documentation: Provide thorough documentation of the system, including user manuals, technical specifications, and developer guides.
- Compliance with Standards: Ensure compliance with relevant industry standards and regulations, particularly those related to privacy, data security, and ethical AI use.

7. Continuous Improvement and Testing

- Iterative Testing: Adopt an iterative approach to testing, continuously evaluating and refining the system based on user feedback and performance metrics.

By adhering to these design standards, your pose estimation project can achieve high levels of accuracy, efficiency, and user satisfaction, ensuring its success in various applications.

5.2 Coding Standards

1. Code Structure and Organization

- Modular Design: Organize the code into logical modules or packages based on functionality (e.g., data preprocessing, model training, inference, etc.).
- Reusable Components: Encourage the use of reusable components and functions to reduce redundancy and facilitate easier updates.

2. Naming Conventions

- Descriptive Names: Use descriptive variable, function, and class names that reflect their purpose or the data they represent.
- Case Conventions: Adopt a consistent case convention (e.g., camelCase for variables and functions, PascalCase for classes) throughout the project.

3. Commenting and Documentation

- In-line Comments: Use in-line comments sparingly to explain "why" behind non-obvious code blocks or decisions.

Docstrings: Use docstrings for every function, method, and class to describe their purpose, parameters, and return types.

- README and Documentation: Maintain a comprehensive README file and detailed documentation, especially for setting up the environment, dataset preparation, and how to run the code.

4. Consistent Coding Style

- Style Guide Adherence: Follow a recognized style guide (e.g., PEP 8 for Python) to ensure consistency across the codebase.

5. Error Handling

- Explicit Error Handling: Use explicit and anticipatory error handling to manage exceptions and unexpected inputs, ensuring the program's robustness.

6. Testing

- Unit Tests: Write unit tests for critical components and functions to ensure they perform as expected.
- Integration Tests: Implement integration tests to verify that different parts of the application work together correctly.

8. Security and Data Protection

- Secure Coding Practices: Follow secure coding practices to protect against vulnerabilities (e.g., data leaks, unauthorized access).
- Sensitive Data Handling: Ensure sensitive data, such as personal information in datasets, is handled according to privacy laws and ethical guidelines.

9. Performance Optimization

- Efficiency in Code: Optimize algorithms and data structures for better performance, especially critical in real-time pose estimation applications.
- Profiling and Optimization Tools: Utilize profiling tools to identify and optimize performance bottlenecks.

10. Code Reviews

- Code Reviews: Conduct regular code reviews to maintain code quality, share knowledge among team members, and catch issues early.

By following these coding standards, your pose estimation project can achieve high levels of technical excellence, maintainability, and team productivity.

5.3 Testing Standards

Incorporating rigorous testing standards is pivotal for the success and reliability of a pose estimation project. Testing ensures that the system performs accurately under various conditions and can handle real-world applications effectively. Below is a detailed outline of testing standards and practices tailored for a pose estimation project, focusing on ensuring functionality, reliability, performance, and user satisfaction.

1. Unit Testing

- Objective: Test individual components or functions in isolation to ensure they perform as expected.
- Standards:
    - Write tests for all new code and critical functions.
    - Use a consistent framework (e.g., pytest for Python) across the project.
    - Aim for a high code coverage percentage to ensure thorough testing.

2. Integration Testing

- Objective: Verify that different modules or services work together as intended.
- Standards:
    - Test the interaction between the data preprocessing modules, the pose estimation model, and the output processing.
    - Use real-world scenarios to ensure that the system components integrate well in practice.

3. System Testing

- Objective: Evaluate the complete and integrated software product to ensure it complies with the specified requirements.
- Standards:
    - Conduct end-to-end testing scenarios that mimic real-world use cases.
    - Include tests for all user flows and failure modes.

4. Performance Testing

- Objective: Ensure that the pose estimation system meets performance criteria under expected and stress conditions.

- Standards:
  - Test the system's performance in terms of speed, memory usage, and processing power both in controlled environments and in real-world conditions.
  - Use profiling tools to identify and optimize bottlenecks.

5. Usability Testing

- Objective: Evaluate the system from the user's perspective to ensure it is intuitive, efficient, and satisfying to use.
- Standards:
  - Involve real users in testing to gather feedback on the system's interface and overall user experience.
  - Test for accessibility and inclusivity to ensure the system is usable by a wide range of people.

6. Security Testing

- Objective: Identify vulnerabilities in the system that could lead to loss of information, information leakage, or unauthorized access.
- Standards:
  - Conduct vulnerability assessments and penetration testing.
  - Ensure data protection and compliance with relevant privacy laws and guidelines.

8. Cross-Platform and Cross-Environment Testing

- Objective: Ensure that the pose estimation system performs consistently across different platforms and environments.
- Standards:
  - Test the system on all supported hardware and software configurations.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Our project has successfully developed a real-time pose estimation model that accurately identifies and tracks human body poses in live video streams. Through the integration of computer vision techniques and machine learning algorithms, we have achieved a system capable of detecting key points on the human body and estimating their relationships in real-time. The significance of our work lies in its potential applications across a wide range of domains, including sports analysis, health-care monitoring, gesture recognition, and human-computer interaction.

6.2 Future Scope

The future scope of our real-time pose estimation project is promising, with numerous opportunities for expansion, refinement, and application in various fields. This system opens up new avenues for innovation in biometric authentication systems, offering improved performance and reliability. Additionally, ongoing research into advanced machine learning algorithms and sensor technologies will continue to drive the evolution of facial recognition systems, paving the way for even more sophisticated and accurate solutions in the future.

Our body language decoder system represents a significant step forward in the realm of human-centered technology, bridging the gap between human communication and artificial intelligence. We are excited about the potential of this technology to transform how we interact with machines and each other, and we look forward to seeing its continued evolution and adoption in the years to come.

References

[1] E Samkari, M Arif, M Alghamdi, MA Al Ghamdi, “Human Pose Estimation Using Deep Learning: A Systematic Literature Review”, Umm Al-Qura University, Makkah 21955, Saudi Arabia, 2023
DOI:10.3390/make5040081

[2] Jyoti Jangade; Kanojia Sindhuben Babulal, “Study on Deep Learning Models for Human Pose Estimation and its Real Time Application”, Central University of Jharkhand Ranchi, India, 2023
DOI: 10.1109/ISCON57294.2023.10112004

[3] Gongjin Lan; Yu Wu; Fei Hu; Qi Hao, “Vision-Based Human Pose Estimation via Deep Learning: A Survey”, vol. 53, China, 2022.
DOI:10.1109/THMS.2022.3219242

[4] Beanbonyka Rim, Nak-Jun Sung, Jun Ma, Yoo-Joo Choi, Min Hong, “Real-time Human Pose Estimation using RGB-D images and Deep Learning”, Korea, 2020

[5] Miniar Ben Gamra, Moulay A. Akhloufi, “A review of deep learning techniques for 2D and 3D human pose estimation”, Dept of Computer Science, Université de Moncton, Moncton, NB, E1A 3E9, Canada, 2021.
DOI: https://doi.org/10.1016/j.imavis.2021.104282

[6] (2019) The Nanonets website.
Available:https://nanonets.com/blog/human-pose-estimation-2d-uide/#deeppose

[7] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, vol.2, 2015, Department of Compuuter Science, University of Toronto, Ontario, Canada.
Available: https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

[8] Pranjal Kumar, Siddhartha Chauhan & Lalit Kumar Awasthi, “Human pose estimation using deep learning: review, methodologies, progress and future research directions”, National Institute of Technology Hamirpur, Hamirpur, Himachal Pradesh, India, 2022.
DOI: https://doi.org/10.1007/s13735-022-00261-6

[9] Jupalle Hruthika, Pulipati Krishna Chaitanya, Goli Shiva Chaithanya, vol. 7 no. 12, “Deep Learning Based Human Pose Estimation Using Opencv”, India, 2020
Available: https://repo.ijiert.org/index.php/ijiert/article/view/2916

INDIVIDUAL CONTRIBUTION REPORT:

POSE ESTIMATION IN REALTIME

PRASHANTA RAJON BAROOAH

21053343

Abstract: This project presents a real-time system that combines Media-pipe libraries and Open-CV to perform pose estimation and facial expression recognition simultaneously. The system accurately tracks key body points and recognizes various facial expressions, enabling applications in fitness tracking, gesture-based interfaces, affective computing, and human-computer interaction.

Individual contribution and findings: I served as a developer responsible for model development in the project on pose estimation in real-time using Media-pipe and OpenCV. My primary role involved designing and fine-tuning the machine learning model architecture for accurate pose estimation.

Overall, my contribution to the project involved thorough research, meticulous planning, and hands-on development of the pose estimation model using MediaPipe and OpenCV. Through this experience, I acquired valuable technical skills and insights into the practical implementation of machine learning techniques in real-world applications.

Individual contribution to project report preparation: I was primarily responsible for crafting Chapter 6 of the project report, which includes the conclusion and future scope sections. Additionally, I undertook the task of gathering and citing references used throughout the report.

Individual contribution for project presentation and demonstration: Played a pivotal role in both preparing the project presentation and conducting the demonstration, focusing specifically on showcasing the conclusion and future scope of the model, as well as demonstrating its real-time functionality using a webcam.

Full Signature of Supervisor: ……………………………. Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

POSE ESTIMATION IN REALTIME

MAHARNAV KASHYAP

2105127

Abstract: This project presents a real-time system that combines Media-pipe libraries and Open-CV to perform pose estimation and facial expression recognition simultaneously. The system accurately tracks key body points and recognizes various facial expressions, enabling applications in fitness tracking, gesture-based interfaces, affective computing, and human-computer interaction.

Individual contribution and findings: I served as a developer responsible for model development. I conducted extensive research on pose estimation techniques, focusing on the integration of MediaPipe and OpenCV for real-time processing. I analyzed various approaches and determined the most suitable methods for our project goals. I collaborated with the data processing team to ensure compatibility between the input data format and the model requirements.

Working within a project group provided invaluable experience in collaboration and communication.Through regular meetings and progress updates sessions, I acquired valuable skills both technical and non-technical.

Individual contribution to project report preparation: In the preparation of the group project report, I undertook the responsibility for crafting the Introduction and Basic Concepts section, which constitutes Chapter 1 and 2 of the report. I conducted thorough research on the topic to gather relevant information and understand the foundation concepts necessary for the project.

Individual contribution for project presentation and demonstration: During the project presentation and demonstration, I took on the responsibility of delivering the introductory segment, which covered the foundation concepts outlined in Chapter 1 and 2 of our project report. Additionally, I demonstrated the basics utilized in the model developed using MediaPipe and OpenCV libraries.

Full Signature of Supervisor: ……………………………….

Full signature of the student: ………………………………..

INDIVIDUAL CONTRIBUTION REPORT:

POSE ESTIMATION IN REALTIME

DEBAPRATIM PAUL

2105116

Abstract: This project presents a real-time system that combines Media-pipe libraries and Open-CV to perform pose estimation and facial expression recognition simultaneously. The system accurately tracks key body points and recognizes various facial expressions, enabling applications in fitness tracking, gesture-based interfaces, affective computing, and human-computer interaction.

Individual contribution and findings: I played a crucial role in the preparation of the group project report by contributing in Data Collection and Dataset Creation. My specific contribution focused on detailing the process and methodology involved in creating the dataset used for training and evaluating the pose estimation model. My role was crucial in laying down the foundation for our project on real-time pose estimation using MediaPipe and OpenCV. In terms of planning, I outlined the steps required to tackle each aspect of the project, ensuring a systematic approach towards achieving our goals. This involved breaking down the problem statement into manageable tasks and allocating resources effectively.

Throughout the implementation phase, I encountered various technical challenges and made several insightful findings. Experimenting with different configurations and parameters, I gained valuable experience.

Individual contribution to project report preparation: As a member of the project group, my primary responsibility revolved around crafting the problem statement, project planning, Software Requirements Specification (SRS), system design, and developing the block diagram of the model implemented which covers chapter 3.

Individual contribution for project presentation and demonstration: In the project presentation and demonstration, I was responsible for presenting the problem statement, which encompassed project planning, the Software Requirements Specification (SRS) document, and the demonstration of block diagrams used in the model.

Full Signature of Supervisor: …………………………….

Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

POSE ESTIMATION IN REALTIME

KUMAR GAURAV

2105125

Abstract: This project presents a real-time system that combines Media-pipe libraries and Open-CV to perform pose estimation and facial expression recognition simultaneously. The system accurately tracks key body points and recognizes various facial expressions, enabling applications in fitness tracking, gesture-based interfaces, affective computing, and human-computer interaction.

Individual contribution and findings: I played a crucial role in translating the proposed methodology into executable code. My planning involved breaking down the methodology into smaller tasks, identifying dependencies, and allocating time effectively to ensure timely completion.

Overall, my contribution to the project involved implementation of the proposed methodology, thorough result analysis, and valuable technical insights gained through hands-on experience. This project not only strengthened my skills in computer vision and software development but also provided me with valuable teamwork and collaboration experience.

Individual contribution to project report preparation: In this report, my individual contribution was primarily focused on implementing Chapter 4: Methodology, Result Analysis, and Testing. This involved significant planning and execution to ensure the successful implementation of the proposed methodology

Individual contribution for project presentation and demonstration: In the project presentation and demonstration, I played a pivotal role in showcasing the implementation aspects of our project, focusing on methodologies, testing, and result analysis which highlighted the approach undertaken in implementing and evaluating our pose estimation model

Full Signature of Supervisor: ……………………………. Full signature of the student: ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

POSE ESTIMATION IN REALTIME

SAMBIT BHATTACHARJEE

2105655

Abstract: This project presents a real-time system that combines Media-pipe libraries and Open-CV to perform pose estimation and facial expression recognition simultaneously. The system accurately tracks key body points and recognizes various facial expressions, enabling applications in fitness tracking, gesture-based interfaces, affective computing, and human-computer interaction.

Individual contribution and findings: In this project, my primary responsibility was the creation of the dataset used for training the pose estimation model. I conducted thorough research on existing datasets for pose estimation to understand their structure, quality, and relevance to our project goals. Based on this research, I formulated a plan for creating our own dataset that would meet our specific requirements. I realized the importance of diversity in the dataset to improve the model's generalization capability.

Overall, my contribution to the project through dataset creation provided a foundation for training the pose estimation model, and the technical insights gained during this process significantly enhanced my understanding of computer vision techniques and their practical application.

Individual contribution to project report preparation: I contributed to the preparation of the project report by focusing on the chapter 5 which was related to the design, coding, and testing standards adopted. Specifically, I was responsible for drafting the sections outlining the design principles, coding practices, and testing methodologies employed throughout the project.

Individual contribution for project presentation and demonstration: For the project presentation and demonstration, my specific role was to showcase the standards adopted in the design, coding, and testing of the model prepared using MediaPipe and OpenCV libraries. By emphasizing these standards, I aimed to convey the professionalism, reliability, and effectiveness of our project implementation.

Full Signature of Supervisor:	Full signature of the student:
…………………………….	……………………………..

Pose Estimation in Real Time

ORIGINALITY REPORT

20%	17%	5%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijariit.com Internet Source	3%
2	Submitted to KIIT University Student Paper	2%
3	www.coursehero.com Internet Source	2%
4	Submitted to Banaras Hindu University Student Paper	2%
5	www.nature.com Internet Source	1%
6	Submitted to nith Student Paper	1%
7	Chengang Dong, Guodong Du. "An enhanced real-time human pose estimation method based on modified YOLOv8 framework", Scientific Reports, 2024 Publication	1%
8	medium.com Internet Source	1%

9	www.google.com Internet Source	1%
10	Submitted to Manipal University Student Paper	1%
11	Submitted to Roehampton University Student Paper	1%
12	developers.google.cn Internet Source	1%
13	Submitted to The Independent Institute of Education (IIE) Student Paper	<1%
14	www.visma.lt Internet Source	<1%
15	wrap.warwick.ac.uk Internet Source	<1%
16	fau.digital.flvc.org Internet Source	<1%
17	Submitted to University of West London Student Paper	<1%
18	vdocuments.mx Internet Source	<1%
19	Arunava Mukhopadhyay, Aritra Chakrabarty, Agnish Arpan Das, Aishik Sarkar. "Hand Gesture Based Recognition System", 2023 7th	<1%

International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), 2023
Publication

20	www.jetir.org Internet Source		<1 %
21	www.researchgate.net Internet Source		<1 %
22	assets.researchsquare.com Internet Source		<1 %
23	docs.microsoft.com Internet Source		<1 %
24	www.hackread.com Internet Source		<1 %
25	www.publications.scrs.in Internet Source		<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		