Human Pose Estimation using Machine Learning

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Honey Tiwari, honey.2226it1149@kiet.edu

Under the Guidance of

P. Raja

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to TechSaksham and AICTE for providing me with this wonderful opportunity to work on the project "Human Pose Estimation using Machine Learning".

I extend my heartfelt thanks to my mentor, whose continuous guidance, support, and encouragement helped me successfully complete this project. Their valuable insights and constructive feedback played a crucial role in refining my approach and improving the overall quality of my work. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

Lastly, I appreciate my peers, family, and friends for their unwavering support and motivation throughout this project journey.

ABSTRACT

Human Pose Estimation is a crucial task in computer vision that involves detecting key points of a human body, such as joints and limbs, in images or videos. It has a wide range of applications, including sports analytics, healthcare, motion tracking, security surveillance, and animation. This project aims to implement a pose estimation system using the **OpenPose deep learning model** and integrate it into an interactive **Streamlit web application**.

The project workflow involves image and video processing using **OpenCV and NumPy**, where input media files are pre-processed before being passed to the **OpenPose model**. The model extracts keypoints, which are then connected to form a skeletal representation of the detected human figure. The application enables users to upload images and videos, view real-time pose detection results, and download processed outputs.

The proposed system is designed to work with single-person and multi-person detection. The thresholding mechanism ensures reliable detection by filtering out low-confidence key points. Additionally, OpenPose's pre-trained model ensures high accuracy in detecting body joints, even in complex poses. Experimental results demonstrate the effectiveness of the system in various scenarios, including static images and dynamic motion sequences. Additionally, the project includes a download feature to save processed outputs.

In conclusion, this project successfully develops an interactive human pose estimation tool. Future enhancements may include real-time webcam-based detection, edge computing optimization for faster inference, and expanded dataset training to improve robustness across diverse environments.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Human Pose Estimation is a rapidly evolving field in computer vision that enables machines to recognize and analyze human body movements by identifying key body joints from images or videos. This technology has significant applications in areas such as sports analytics, healthcare, security surveillance, and animation. This chapter provides an overview of the **problem statement, motivation, objectives, and scope** of the project.

1.1 Problem Statement:

Understanding **human movements and postures** is crucial in various fields, including **healthcare, sports, security, and human-computer interaction**. Traditionally, analysing human motion required **manual annotation** or specialized motion capture devices, which are expensive, intrusive, and impractical for large-scale applications. **Human Pose Estimation** aims to solve this problem by using **computer vision and deep learning** to automatically detect key body joints from images or videos. This enables applications such as **motion analysis, activity recognition, rehabilitation monitoring, virtual reality, and robotics**. This project aims to develop a **deep learning-based pose estimation system using OpenPose** to accurately track human movements from images and videos. By integrating this system into a web-based interface, users can easily analyze human poses without the need for specialized hardware, making pose estimation more accessible and practical for real-world applications.

1.2 Motivation:

The motivation behind this project stems from the growing need for **automated human motion analysis** in various fields:

Healthcare & Rehabilitation – Used for patient monitoring, posture correction, and physical therapy.

Sports Analytics – Helps athletes improve performance by analyzing body movements.

Surveillance & Security – Enhances public safety by identifying suspicious activities.

Animation & Gaming – Improves character animation in gaming and virtual reality.

Deep learning-based pose estimation eliminates the need for **expensive motion capture systems** and provides a **cost-effective solution** for real-time motion tracking.

1.3 Objective:

The primary objective of this project is to develop an **AI-driven pose estimation system** that accurately detects human body joints from images and videos using the **OpenPose model**. This system aims to provide an interactive and accessible solution for **motion analysis, activity recognition, and human posture tracking** without requiring specialized hardware. The specific objectives of this project are:

- Develop a deep learning-based pose estimation system using OpenPose.
- Implement a web-based application for easy accessibility.
- Design an intuitive web-based interface using Streamlit.
- Process both images and videos to detect human poses accurately
- Ensure real-time efficiency and high accuracy.
- Incorporate a download feature for further analysis.
- Explore potential real-world applications of pose estimation in fields such as sports, healthcare, security, and animation.

1.4 Scope of the Project:

Human pose estimation has diverse applications, from **motion analysis and healthcare monitoring to sports performance tracking and interactive AI systems**. This project explores the implementation of **computer vision-based pose estimation**, offering a practical solution for analyzing human movements.

1.4.1 Key Features:

1. **Supports both image and video processing,** enabling detailed movement analysis.
2. **Interactive web-based interface** that allows users to upload files and visualize results.
3. **Download functionality** for saving processed outputs.
4. **Scalable design,** allowing potential enhancements such as real-time tracking and multi-person detection.

1.4.2 Limitations:

1. Not optimized for **multi-person tracking** in its current version.
2. Requires **sufficient computational power (GPU recommended)** for large video processing.
3. Performance may vary in **low-light conditions or occluded scenarios**.

This project serves as a **foundation for further advancements**, including improved efficiency, gesture recognition, and enhanced real-time performance.

CHAPTER 2

Literature Survey

This chapter reviews existing literature and previous research in the field of human pose estimation (HPE). It explores different methodologies, models, and techniques used for pose estimation, highlighting their strengths and limitations. Additionally, this chapter identifies research gaps and explains how the proposed system addresses these shortcomings.

2.1 Review relevant literature:

Human Pose Estimation (HPE) has been an active area of research in computer vision and deep learning. Over the years, various approaches like traditional computer-vision based techniques and deep-learning based methods have been developed to track and analyze human body movements using image processing and machine learning techniques.

2.1.1 Traditional Approaches:

Earlier pose estimation methods relied on **handcrafted features** and **statistical models**, including:

- **Histogram of Oriented Gradients (HOG) + Support Vector Machines (SVM):** This method detects body parts based on edge orientations, but it struggles with occlusions and complex poses.
- **Deformable Part Models (DPM):** Uses a probabilistic framework to detect body parts, but it requires extensive manual feature engineering.
- **Graphical Models:** Represents the human body as a connected set of parts, optimizing pose estimation through energy minimization techniques.

Despite their contributions, these methods are **computationally expensive** and lack generalization, making them unsuitable for real-time applications.

2.1.2 Deep-Learning Based Approaches:

With advancements in deep learning, Convolutional Neural Networks (CNNs) and neural architectures have significantly improved the accuracy and efficiency of pose estimation. Some notable models include:

- **OpenPose:** A bottom-up multi-person pose estimator that detects keypoints for multiple people in an image, leveraging **Part Affinity Fields (PAFs)** to associate body parts.
- **PoseNet:** A single-shot keypoint detection model that is efficient but less precise in complex poses.

- **HRNet (High-Resolution Network):** Maintains high-resolution feature maps throughout the network, achieving state-of-the-art accuracy.
- **DeepPose (by Google):** Treats pose estimation as a regression problem, directly predicting joint coordinates from images.
- **AlphaPose:** A top-down approach that first detects persons using an object detector and then applies pose estimation for each detected region.
- **DensePose:** Goes beyond keypoint detection to map pixels to 3D human body models. It is used for augmented reality (AR) and gaming applications.
- **BlazePose:** Designed for real-time, mobile-friendly pose estimation. It is used in fitness applications and gesture recognitions.

These deep learning models have revolutionized pose estimation by **eliminating manual feature extraction**, improving accuracy, and enabling real-time processing. However, challenges such as **occlusions, complex movements, and computational costs** still exist.

Model	**Approach**	**Strength**	**Weakness**
OpenPose	Bottom-up	Real-time, multi-person	High computational cost
PoseNet	Single-shot	Mobile-friendly	Lower accuracy
HRNet	High-resolution CNN	State-of-the-art accuracy	Slow inference speed
DeepPose	Regression-based	Refinement improves accuracy	Struggles with occlusion
AlphaPose	Top-down	High precision	Dependent on person detection
DensePose	Dense regression	3D mapping	Requires more computational power
BlazePose	Lightweight CNN	Efficient, real-time	Limited to specific use cases

Table-1: Deep-Learning Based Approaches

2.2 Existing Models, Technologies, and Methodologies:

Pose estimation models can be classified based on various factors such as input type, processing method, and output representation.

2.2.1 Types of Pose Estimation Models:

A. **Single-Person Pose Estimation:** Detects keypoints for a single individual per image. Example: **PoseNet, DeepPose, HRNet**
B. **Multi-Person Pose Estimation:** Detects keypoints for multiple individuals in the same image/video. Example: **OpenPose, AlphaPose**
C. **2D Pose Estimation:** Detects keypoints in 2D image space. Example: OpenPose
D. **3D Pose Estimation:** Predicts keypoints in 3D space using depth information. Example: VIBE, HMR
E. **Supervised Learning Approaches:** Requires labeled datasets. Example: COCO, MPII
F. **Unsupervised Learning Approaches:** Learns pose representations without labeled data.

2.2.2 Technologies and Frameworks Used in Pose Estimation:

Technology	Usage
TensorFlow/ PyTorch	Deep learning frameworks for model training and inference
OpenCV	Preprocessing images/videos before pose estimation
MediaPipe	Lightweight, real-time pose estimation framework
Streamlit	Web-based UI for interactive applications
CUDA&TensorRT	GPU acceleration for real-time inference

Table-2: Technologies & Frameworks

2.3 Research Gaps and Limitations in Existing Solutions:

Despite significant progress, existing pose estimation models face several challenges:

Limitation	Cause	Proposed Solution
Occlusion Handling	People blocking each other in crowded scenes.	Use temporal context from previous frames.
Real-Time Performance	High accuracy models require heavy computation.	Optimize models for edge devices using TensorFlow Lite.
Generalization to Unseen Poses	Most models are trained on fixed datasets.	Use domain adaptation techniques.
Motion Blur in Videos	Fast movements cause loss of detail.	Implement frame interpolation and optical flow-based methods.

Table-3: Limitations in Existing Solutions

2.3.1 How Our Project Addresses These Gaps:

Our system improves upon existing solutions by:

- Combining deep learning models (e.g., OpenPose) with real-time processing for better occlusion handling.
- Enhancing computational efficiency using optimized deep learning frameworks.
- Providing a user-friendly web interface with image and video processing support.
- Leveraging OpenCV and TensorFlow for efficient video processing, making it suitable for healthcare, sports, and fitness tracking.

2.4 Conclusion:

This chapter reviewed existing literature, methodologies, and technologies in human pose estimation. It highlighted traditional and modern approaches, including deep learning models, and identified the limitations of current solutions.

CHAPTER 3

Proposed Methodology

This project aims to implement a pose estimation system using the **OpenPose deep learning model** and integrate it into an interactive **Streamlit web application**. The project workflow involves image and video processing using **OpenCV and NumPy**, where input media files are pre-processed before being passed to the **OpenPose model**. The system follows a **deep learning-based approach** to detect and analyze human poses from images and videos. This chapter provides a detailed overview of the proposed methodology for human pose estimation, covering the system design, workflow, and required tools and technologies.

3.1 System Design:

The system follows a deep learning-based approach to detect and analyze human poses from images and videos. The overall workflow consists of the following steps:

- **User Input:** The user uploads an image or video via the **Streamlit-based web interface**.
- **Preprocessing:** The uploaded media undergoes **image resizing, normalization, and conversion** using OpenCV to ensure compatibility with the pose estimation model used.
- **Pose Estimation Model:** A **pre-trained deep learning model** (such as OpenPose or PoseNet) is used to extract **human body keypoints** which are then connected to form a skeletal representation of the detected human figure.
- **Post-Processing & Visualization:** The detected keypoints are overlaid on the original image or video, forming a **pose visualization** which is shown as output on the web app.
- **Output Display & Download:** The processed image or video is displayed on the web app, with an option to **download** the results in the system compatible format so that user can view the visualized image later on too.

3.1.1 System Architecture:

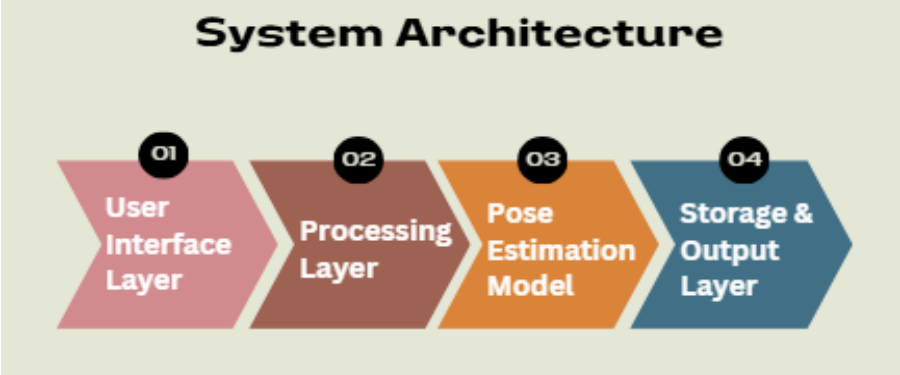

Figure-1: System Architecture

Diagram Explanation:

- **User Interface Layer:** Provides an intuitive interface for uploading media and viewing results (Streamlit-based).
- **Processing Layer:** Handles input validation, preprocessing, and execution of the pose estimation model.
- **Pose Estimation Model:** Uses deep learning (OpenPose/PoseNet) to detect keypoints and draw skeletal connections.
- **Storage & Output Layer:** Temporarily stores processed images/videos and allows users to download them.

3.2 Requirement Specification:

3.2.1 Hardware Requirements:

Component	Specification	Reason
Processor	Intel i5/i7 or AMD equivalent	Ensures smooth model execution.
RAM	Minimum 8GB (16GB recommended)	Handles high computational load.
Storage	Minimum 10GB free space	Stores processed media and models.
GPU(Optional)	NVIDIA GPU (CUDA-enabled)	Enhances deep learning performance.

Table-4: Hardware Requirements

3.2.2 Software Requirements:

Software	Version	Purpose
Python	3.8+	Core Programming Language
OpenCV	Latest	Image & video processing
TensorFlow	Latest	Deep learning model execution
Streamlit	Latest	Web-based UI framework
NumPy	Latest	Numerical computations
Pillow	Latest	Image manipulation

Table-5: Software Requirements

3.2.3 Development Tools:

IDE: VS Code, Jupyter Notebook or PyCharm

Libraries: Matplotlib (for visualizations), TempFile (for handling temporary files), PIL or Python Imaging Library (for opening, manipulating and saving many different image file formats), base64 (for encoding and decoding data for downloading and storing images)

CHAPTER 4

Implementation and Result

This chapter describes the implementation details of the pose estimation system, including the workflow, tools used, code structure, and final results.

4.1 Implementation Details:

The project was implemented using the following technologies:

- **Programming Language:** Python
- **Deep Learning Framework:** OpenPose (Pre-trained Model)
- **Libraries Used:**
  a. OpenCV – Image & video processing
  b. Streamlit – Web-based user interface
  c. NumPy – Numerical computations
  d. TensorFlow – Model execution
- **Development Environment:** VS Code
- **Version Control:** Git & GitHub

4.1.1 Workflow:

1. **User uploads an image or video** through the Streamlit web app.
2. The **OpenPose model** processes the input to detect human body keypoints.
3. The **detected keypoints are visualized** as skeleton overlays on the original image/video.
4. The **processed output is displayed** in the web app.
5. Users can **download** the processed image or video.

4.2 Code Structure:

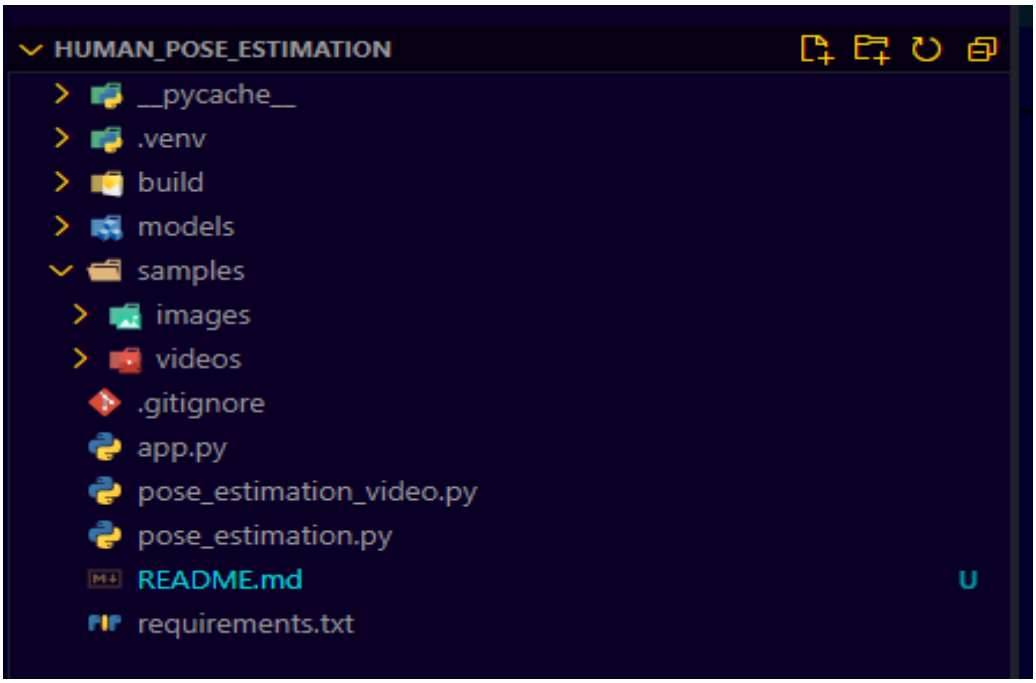

Figure-2: Code Structure

4.3 Snapshots of Results:

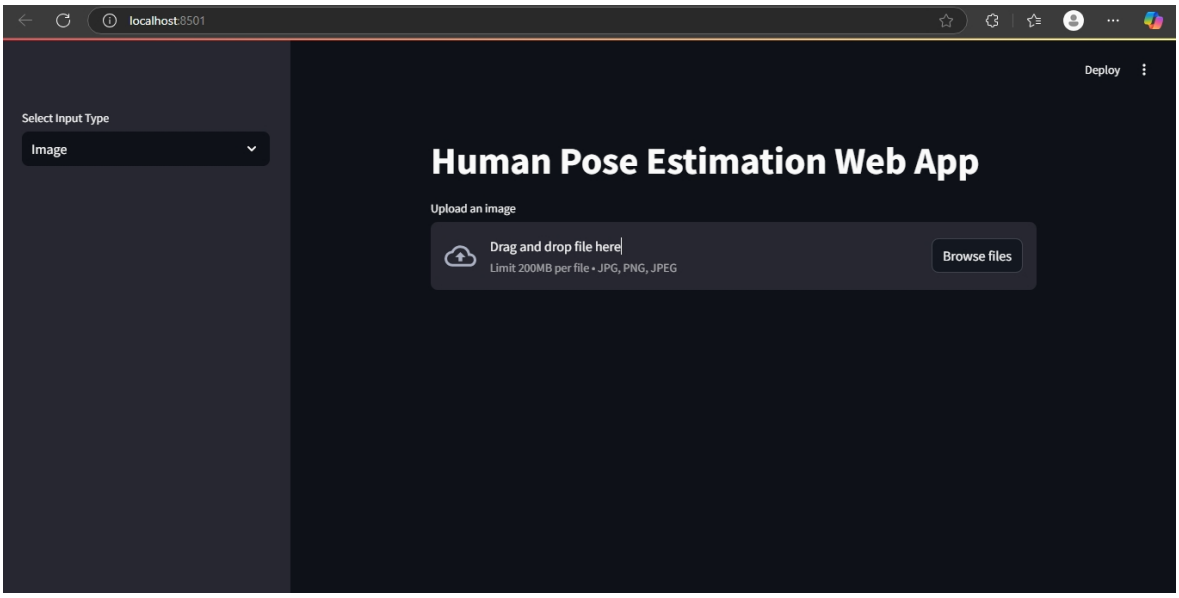

Figure-3: App Page

This figure shows the output window that opens on running the command "streamlit run app.py"

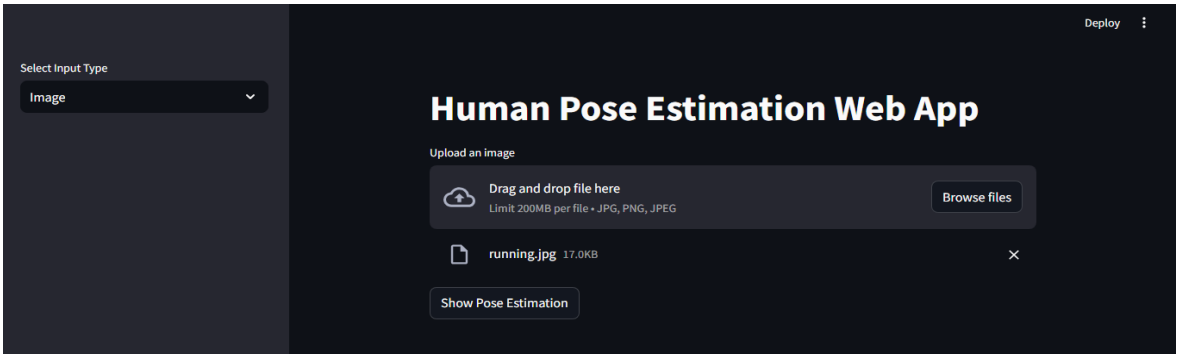

Figure-4: Image Upload Page

This page shows the image that has been uploaded by the user.

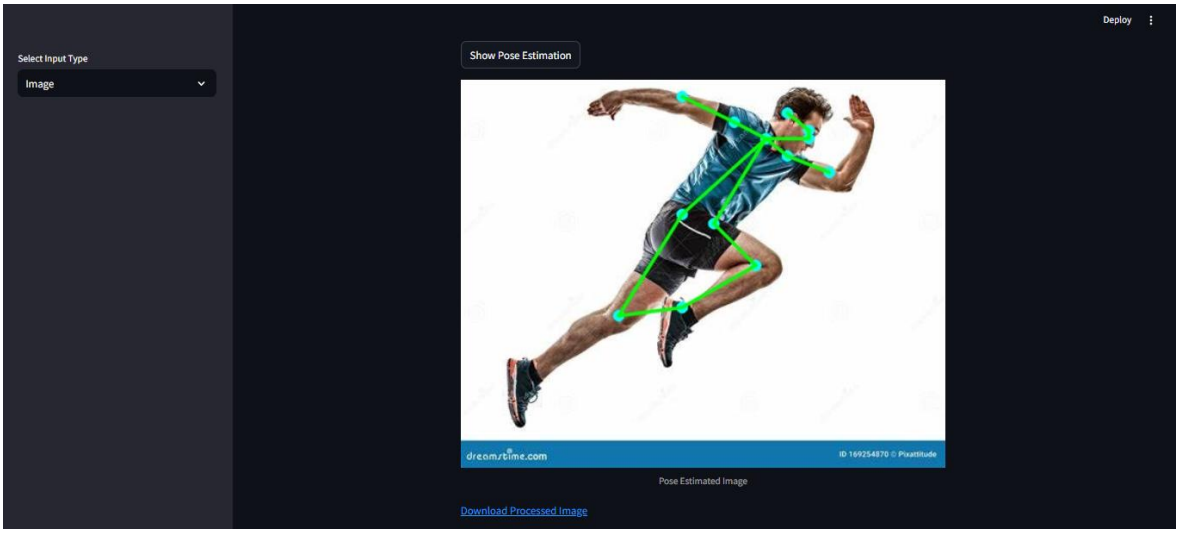

Figure-5: Pose Estimated Page

On clicking the button “Show Pose Estimation”, the estimation gets generated and the output is displayed on the app. On clicking Download Processed image the image gets downloaded as ‘pose_estimated.format’.

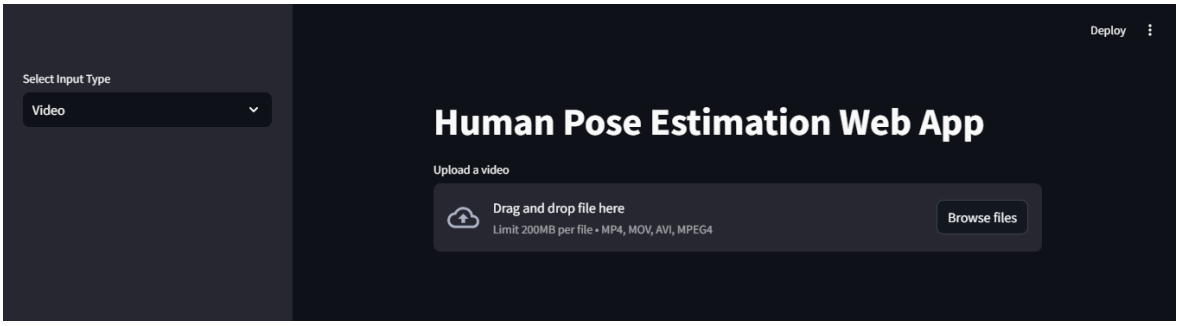

Figure-6: Video Upload Page

This page shows the similar setup for the videos page.

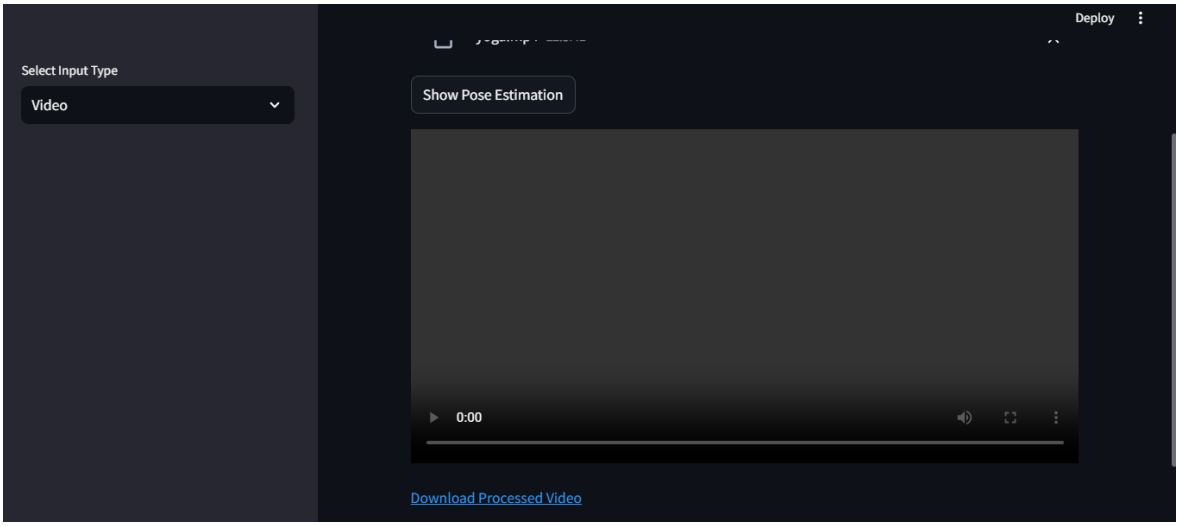

Figure-7: Pose Estimated from Video Page

This image shows the video output generated on clicking the 'Show Pose Estimation' button. To save the processed video we click on the 'Download Processed Video'.

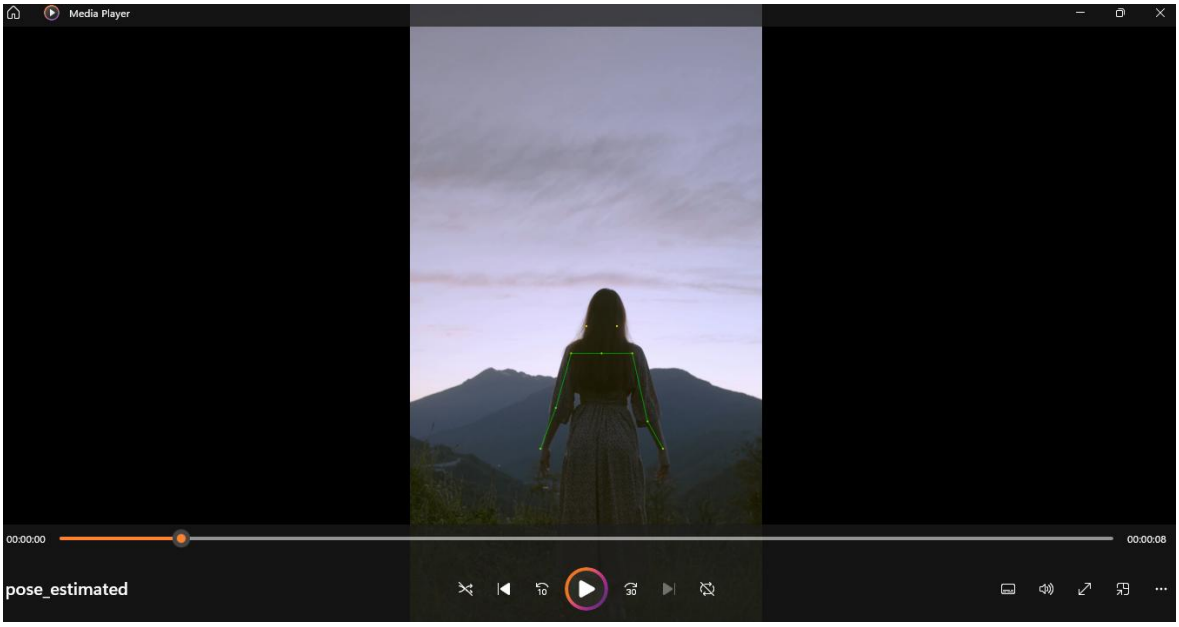

Figure-8: Downloaded Video

This shows a particular instance from the video that has been processed by the model and then downloaded.

4.4 GitHub Link for Code:

The complete source code is available on GitHub:

🔗 **GitHub Repository Link**

CHAPTER 5

Discussion and Conclusion

This chapter presents the key findings, limitations, and possible future improvements for the Human Pose Estimation system. It also summarizes the overall contributions and impact of this project.

5.1 Future Work:

Despite achieving significant progress in real-time **pose estimation using deep learning**, there are still areas for **improvement and expansion**. The following are key directions for future research and development:

a. Improving Model Accuracy:

- Handling Occlusion Better-> Current models struggle when body parts are occluded (e.g., a person blocking another in a crowded scene).
- Pose Estimation in Low-Light Conditions-> Performance degrades in poor lighting or blurry images. Implementing low-light image enhancement techniques and self-supervised learning can improve robustness.

b. Enhancing Real-Time Performance:

- Optimizing for Edge Devices-> Current deep learning models are computationally expensive. Future work can focus on **lightweight models** like MobilePose, or quantizing models using **TensorFlow Lite** for mobile deployment.
- Reducing Latency for Video Processing-> High-resolution video processing is computationally intensive. Implementing **CUDA-accelerated inference** and **multi-threading techniques** can reduce **frame drop issues**.

c. Extending the Application Scope:

- **Expanding to 3D Pose Estimation->** The current model focuses on 2D pose estimation. Future versions can incorporate **3D reconstruction** models (e.g., HMR, VIBE) to estimate depth and improve accuracy.

- **Integration with AR/VR for Interactive Applications**-> Pose estimation can be used in **virtual fitness training, motion gaming, and physiotherapy.**
- **Human Activity Recognition (HAR)->** Pose estimation can be extended for **action recognition** (e.g., identifying exercises, dance movements, or fall detection in healthcare).

5.2 Conclusion:

This project successfully implemented **human pose estimation** using deep learning and computer vision techniques. The system **detects and tracks key human joints** in real time, demonstrating applications in **fitness tracking, motion analysis, and healthcare monitoring**. The project made the following contributions:

- **Implemented a deep learning-based pose estimation system** using TensorFlow and OpenCV.
- **Achieved real-time performance**, making it suitable for **real-world applications**.
- **Provided a user-friendly interface** using Streamlit, making pose analysis accessible to non-technical users.
- **Explored potential improvements**, identifying future work for **enhancing accuracy, efficiency, and real-world usability**.

Despite some **challenges**, such as **occlusion handling and computational complexity**, this project lays a strong foundation for **future enhancements**, including **3D pose estimation, AR/VR applications, and activity recognition**.

In conclusion, **this project contributes to advancing real-time pose estimation**, making it valuable for **fitness, healthcare, and interactive AI applications**. Future work will focus on **improving model robustness, expanding datasets, and optimizing real-time inference** for broader deployment.

REFERENCES

Research Papers & Journals:

- A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, Oct. 2016, pp. 483–499. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46484-8_29
- Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, Jan. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8998684
- Papers with Code, "Human Pose Estimation Models," [Online]. Available: https://paperswithcode.com/task/pose-estimation

Technologies & Frameworks:

- **OpenPose - CMU Perceptual Computing Lab**, "OpenPose: Open-Source Realtime Multi-Person Pose Estimation," [Online]. Available: https://github.com/CMU-Perceptual-Computing-Lab/openpose
- **OpenCV,** "Open Source Computer Vision Library," [Online]. Available: https://opencv.org/
- **Streamlit,** "The Fastest Way to Build Data Apps," [Online]. Available: https://streamlit.io/