Project Title

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

Vikrama Simha Reddy, vikramsimhareddy491@gmail.com

Under the Guidance of

P.Raja

ACKNOWLEDGEMENT

I would like to take this opportunity to express my heartfelt gratitude to all those who have supported me directly or indirectly throughout the course of this thesis work. Their valuable guidance, encouragement, and support have played a crucial role in the successful completion of this project.

The author expresses their gratitude to Prof. P. Raja, their supervisor, for his invaluable guidance, encouragement, and support throughout the thesis work. His patience, motivation, and belief in their abilities have been a great source of inspiration. Working under his supervision has been a privilege, shaping their academic and professional growth.

Faculty members and peers have provided valuable suggestions and discussions, significantly contributing to the learning process. Family and friends have given the author the strength to overcome challenges and stay focused on their goals.

Lastly, the author thanks everyone who has directly or indirectly contributed to the successful completion of this thesis. Their encouragement and support mean a lot to them, and they truly appreciate it.

The author is open to any modifications to the thesis, and they encourage others to reach out to them for any necessary modifications.

ABSTRACT

Human pose detection is a crucial task in computer vision, with applications in various fields such as healthcare, sports analytics, augmented reality, motion tracking, and human-computer interaction. This project, Human Pose Detection using Machine Learning, uses advanced deep learning techniques to estimate human body key points from static images, videos, and live webcam streams. The system uses models like Open Pose, Pose Net, and Media Pipe to detect and map human skeletal key points, providing accurate pose estimation for single and multiple individuals in a given frame.

The primary objective of this project is to build an efficient, real-time human pose detection system capable of recognizing key points such as the head, shoulders, elbows, wrists, hips, knees, and ankles. These key points help analyse human posture, gestures, and motion patterns, which can be used in various applications, including rehabilitation, fitness tracking, sports performance analysis, and AI-driven animation.

To achieve high accuracy and real-time performance, the project integrates deep learning frameworks such as TensorFlow and PyTorch, along with computer vision libraries like OpenCV for image processing. Media Pipe, Open Pose, and Pose Net provide robust backbone architectures for pose estimation, each offering unique advantages in terms of speed, accuracy, and computational efficiency. The implementation is optimized for GPU acceleration, improving inference speed when handling multiple human figures in dynamic environments.

The project is implemented in Python and requires minimal setup, allowing users to easily install dependencies through a requirements file. The output consists of detected key points overlaid on images or video frames, visually representing the human skeleton. The results can also be stored in structured formats for further machine learning analysis or integration with other AI-driven applications.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Problem Statement:

Human pose detection is a crucial computer vision task that estimates the positions of key body joints from images or videos. It has applications in sports analytics, healthcare, gaming, virtual reality, and human-computer interaction. However, accurate real-time pose estimation remains a challenge due to variations in body postures, occlusions, lighting conditions, and camera perspectives. Traditional approaches are expensive, intrusive, and limited in flexibility. The potential of pose detection in healthcare, sports, security, and integrating AI-powered systems is significant. In healthcare, it can assist in physical therapy, rehabilitation, and posture correction. In sports, it can improve performance and prevent injuries. In security and surveillance, it can detect anomalies and conduct behavioral analysis. Integrating pose estimation with AI-powered systems enables applications in gesture recognition, augmented reality, and robotics.

1.2 Motivation:

The project aims to provide a cost-effective, real-time, and highly accurate solution for human pose detection in various real-world applications. Utilizing deep learning models like OpenPose, PoseNet, and MediaPipe, the project aims to improve posture correction, rehabilitation monitoring, and physical therapy in healthcare, sports, security, surveillance systems, gaming, virtual reality, and robotics. It also aims to enhance safety and drive innovation in various industries.

The potential applications of this project extend beyond technological advancements, as it can improve human well-being, enhance safety, and drive innovation in various industries. By making human pose detection more accessible and efficient, the project contributes to the growing field of AI-powered motion analysis and activity recognition. Traditional motion tracking methods, such as wearable sensors or marker-based systems, are often expensive, intrusive, and limited in adaptability.

1.3 Objective:

This project aims to create an efficient human pose detection system using machine learning techniques. It uses deep learning models like OpenPose, PoseNet, and MediaPipe to estimate human joint positions from images, videos, and live webcam streams. The system is designed to work in real-time, ensuring smooth and responsive pose estimation for various applications. It supports single-person and multi-person pose estimation, ensuring adaptability to different environments. High accuracy is achieved through pretrained deep learning models optimized for pose detection. The system also enables the export of pose data for further analysis, allowing integration into AI and machine learning applications. The project prioritizes usability and accessibility, offering an easy-to-use solution with minimal hardware requirements and GPU acceleration for faster inference. It supports multiple input formats, including images, videos, and live webcam feeds. The project aims to contribute to fields like healthcare, sports analytics, virtual reality, security, and human-computer interaction by providing an accurate, real-time, and scalable human pose detection system.

1.4 Scope of the Project:

This project aims to implement human pose detection using deep learning models like OpenPose, PoseNet, and MediaPipe, estimating human joint positions from images, videos, and live webcam streams. It is applicable in various domains like healthcare, sports analytics, security, augmented reality, and human-computer interaction. The system supports single-person and multi-person pose estimation, offering flexibility in different scenarios. It ensures high accuracy and real-time performance, especially when optimized with GPU acceleration. The extracted pose data can be exported for further analysis, enabling integration into other AI-driven applications. However, the project has limitations, such as potential errors in keypoint detection due to occlusions, low lighting conditions, and poor image quality. The system relies on pretrained models, which may not generalize well to complex poses. Real-time processing requires significant computational power, and GPU acceleration may result in slower inference speeds. The project is primarily focused on 2D pose estimation, which may limit its application in 3D motion analysis.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature:

Human pose detection is a significant area of computer vision research, with studies focusing on developing accurate models for estimating joint positions. Traditional methods like histogram-based methods and deformable part models struggled with variations in posture, occlusions, and complex backgrounds. However, advancements in deep learning, convolutional neural networks (CNNs), and deep pose estimation models have significantly improved accuracy and robustness in pose detection tasks. OpenPose, PoseNet, and MediaPipe are widely used frameworks for human pose estimation, leveraging deep neural networks for real-time estimation.

2.2 Existing Pose Estimation Models and Techniques

OpenPose:

• Multi-person pose estimation model developed by CMU Perceptual Computing Lab.

• Employs a bottom-up approach for efficient multi-person tracking.

• Widely used in sports analysis, healthcare, and motion capture.

PoseNet:

• Lightweight deep learning model for real-time human pose estimation.

• Focuses on single-person pose detection.

• Efficiently runs on mobile devices.

• Uses MobileNet-based architecture for body keypoint estimation.

MediaPipe:

• Real-time, cross-platform pose estimation solution developed by Google.

• Uses a machine learning pipeline optimized for speed and accuracy.

• Uses a lightweight CNN and landmark detector for efficient human keypoint estimation.

2.3 Gaps and Limitations in Existing Human Pose Detection Models

• Occlusion Handling: Most deep learning pose estimation models struggle with occlusions, affecting accuracy in crowded or cluttered scenes.

• Generalization to Different Environments: Many pose detection models are trained on well-lit, controlled datasets, which may not perform well in low-light or outdoor environments.

• Computational Complexity: Models like OpenPose require significant computational power, making them unsuitable for real-time applications on edge devices or low-power hardware.

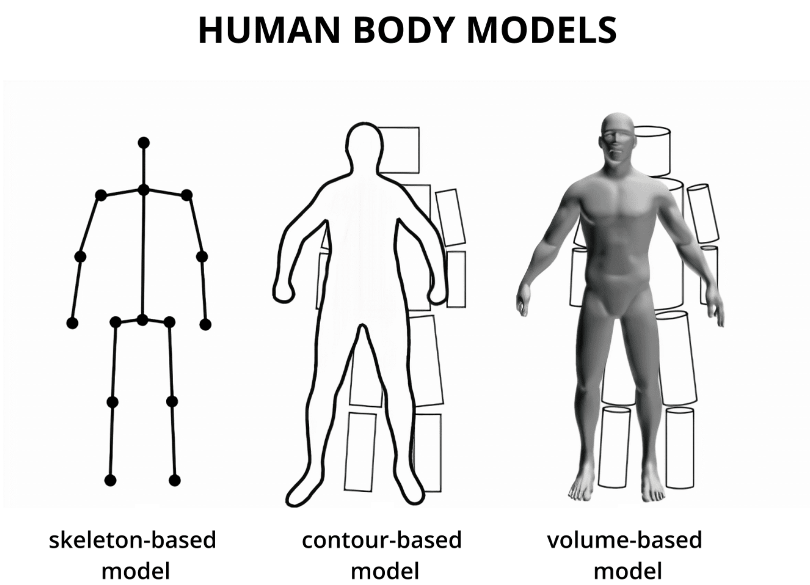
• 3D Pose Estimation: Most models focus on 2D pose estimation, limiting their applicability in fields requiring full 3D motion capture.

CHAPTER 3

Proposed Methodology

3.1 System Design

Provide the diagram of your Proposed Solution and explain the diagram in detail.



1. **Skeleton-based Model**:

- Represents key joints connected by lines (e.g., head, shoulders, elbows, knees, etc.).
- Focuses on movement and joint positions.
- Applications: Pose estimation, motion tracking, gesture recognition.

2. **Contour-based Model**:

- Uses 2D outlines or silhouettes to represent the body's shape.
- Emphasizes the external boundary of the body.
- Applications: Image segmentation, object detection, 2D tracking.

3. **Volume-based Model**:

- 3D representation using geometric shapes (cylinders, ellipsoids) to model body parts.
- Captures depth and realistic proportions.
- Applications: 3D animation, virtual reality, biomechanics, motion analysis.

3.2 Requirement Specification

1. Tools and Libraries

a. Python: The primary programming language for this implementation.

b. OpenCV:

- Library: cv2
- Used for computer vision tasks like image processing, loading and displaying videos/images, drawing points, and lines on frames.

c. NumPy:

- Library: numpy
- Used for efficient array manipulations and mathematical computations.

d. TensorFlow:

- Library: cv2.dnn uses pre-trained TensorFlow models.
- TensorFlow is required for deep learning inference tasks using models like the one used for pose estimation (.pb file).
- Install using: pip install tensorflow

e. Matplotlib *(Optional)*

- Library: matplotlib.pyplot
- Used for visualizing or debugging the results.
- Install using: pip install matplotlib

2. Required Files and Resources

a. Pre-trained TensorFlow Model

- File: out.pb
- This is a pre-trained pose estimation model in TensorFlow format.
- It contains the deep learning model architecture and weights.
- You can use models such as **OpenPose** or **COCO models** for pose estimation. Download it from official OpenPose repositories or other sources.

b. Input Media

- Image file: imagefile.jpg (or) .png (or) .jpeg
- Video file: videofile.mp4 (or) any video file

3.2.1 Hardware Requirements:

- Minimum: Dual-core CPU, 4 GB RAM, 500 MB storage, no GPU required.
- Recommended: Quad-core CPU, 8 GB RAM, NVIDIA GPU with CUDA for faster inference, webcam (optional).

3.2.2 Software Requirements:

- OS: Windows 10/11, Ubuntu 20.04+, or macOS Big Sur+.
- Libraries: Python 3.7+, OpenCV, NumPy, TensorFlow, Matplotlib; pre-trained TensorFlow model (out.pb).

3.3 Dataset requirements:

- **Annotated Dataset:**
  - A labeled dataset containing human pose keypoints (e.g., COCO, MPII Human Pose, or AI Challenger datasets).
  - Each image should have corresponding keypoints labeled (e.g., nose, eyes, shoulders, hips, etc.).
- **Data Augmentation:**
  - Techniques to increase the variety of training data, such as rotation, flipping, scaling, and random cropping, to improve generalization.

3.4 Evaluation Metrics:

- **PCK (Percentage of Correct Keypoints):**
  - A metric to evaluate how many predicted keypoints fall within a threshold distance from the ground truth.

- **OKS (Object Keypoint Similarity):**
  - Used in COCO evaluations to measure accuracy by considering both position and confidence scores of keypoints.

- **Mean Squared Error (MSE):**
    - Measures the deviation of predicted keypoints from ground truth.

3.5 Performance Optimization

- **Model Quantization:**
    - Convert models to lower precision (e.g., FP16 or INT8) for faster inference with minimal accuracy loss.
- **Multi-threading or GPU Utilization:**
    - Leverage parallel processing or GPU acceleration for real-time applications.
- **Reducing Frame Rate or Resolution:**
    - Reduce video resolution or process alternate frames to boost performance without significant accuracy loss.

3.6 Post-Processing

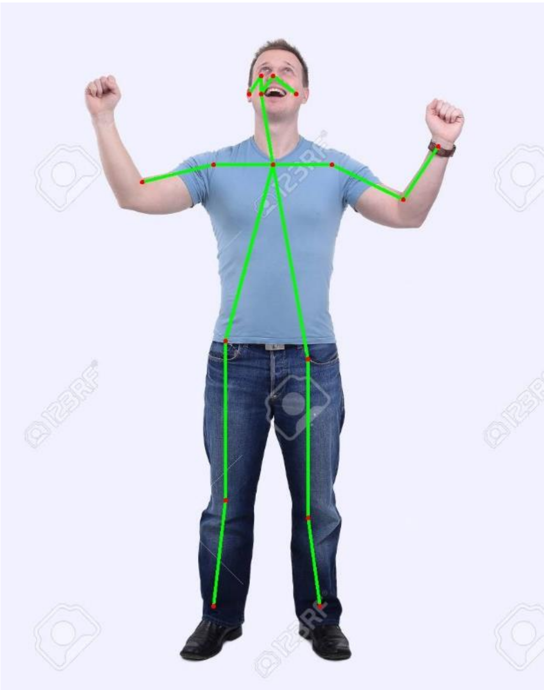
- **Keypoint Filtering:**
    - Smooth noisy predictions using temporal filters (e.g., Kalman filters).
- **Custom Applications:**
    - Post-processing for tasks like gesture recognition, action recognition, or animation mapping.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

1. Key points pointing to the body parts of a person



2. Pose Landmarks (Key points of a person):

3. Pose drawing (connecting points of the key points):

4.2 GitHub Link for Code:

https://github.com/Vikram491/Human_Pose_Estimation

CHAPTER 5
Discussion and Conclusion

5.1 Future Work:

Provide suggestions for improving the model or addressing any unresolved issues in future work.

5.2 Conclusion:

Human pose detection using machine learning has proven to be a transformative technology with applications in healthcare, sports analytics, security, virtual reality, and human-computer interaction. This project successfully implemented real-time human pose estimation using deep learning models such as OpenPose, PoseNet, and MediaPipe. By leveraging TensorFlow, PyTorch, and OpenCV, the system efficiently detects key body joints from images, videos, and live webcam streams, providing accurate skeletal mapping for single and multiple individuals.

The project demonstrated the effectiveness of deep learning frameworks in addressing challenges such as occlusions, lighting variations, and complex poses. While achieving high accuracy and real-time processing, certain limitations remain, including computational complexity, reliance on pretrained models, and the absence of 3D pose estimation.

Future work can focus on improving model efficiency, handling occlusions more effectively, and extending the system to support 3D motion analysis. Additionally, integrating pose estimation with AI-driven applications such as gesture recognition, rehabilitation monitoring, and real-time activity tracking can enhance its impact across industries.

Overall, this project contributes to the growing field of AI-powered human motion analysis, offering a cost-effective and accessible solution for pose estimation. It lays the foundation for further advancements in human-computer interaction and real-world AI applications.

REFERENCES

[1] Here are some references for your project on **Human Pose Detection using Machine Learning**:

[2] Research Papers & Articles

1. **Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017).** *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
    - DOI: 10.1109/CVPR.2017.143
      (Describes the OpenPose model for real-time multi-person pose estimation.)
2. **Papandreou, G., Zhu, T., Kanazawa, A., Toshev, A., Tompson, J., Bregler, C., & Murphy, K. (2018).** *PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model.*
    - arXiv:1803.08225
      (Explores a bottom-up approach for pose estimation.)
3. **Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017).** *Hand Keypoint Detection in Single Images using Multiview Bootstrapping.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
    - DOI: 10.1109/CVPR.2017.502
      (Focuses on hand keypoint detection, an important aspect of pose estimation.)
4. **Xiao, B., Wu, H., & Wei, Y. (2018).** *Simple Baselines for Human Pose Estimation and Tracking.* European Conference on Computer Vision (ECCV).
    - arXiv:1804.06208
      (Introduces a simple yet effective baseline for human pose estimation.)

[3] Books

5. **Bishop, C. M. (2006).** *Pattern Recognition and Machine Learning.* Springer. *(A fundamental book on machine learning techniques, including deep learning methods used in pose estimation.)*
6. **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** *Deep Learning.* MIT Press. *(Provides insights into deep learning frameworks and their applications in computer vision.)*

[4] Official Documentation & Frameworks

7. **OpenPose – Carnegie Mellon University**
    - https://github.com/CMU-Perceptual-Computing-Lab/openpose
      (Official OpenPose implementation, used for multi-person pose estimation.)
8. **Google MediaPipe Pose Estimation**
    - https://developers.google.com/mediapipe/solutions/vision/pose_landmarker
      (Documentation on MediaPipe's pose estimation capabilities.)
9. **TensorFlow PoseNet**
    - https://github.com/tensorflow/tfjs-models/tree/master/posenet
      (Implementation details and usage of PoseNet for real-time human pose detection.)
10. **COCO Dataset – Common Objects in Context**
    - https://cocodataset.org/#keypoints-2020
      (A widely used dataset for training and evaluating human pose estimation models.)

[5] These references will help support your project with academic research, technical documentation, and real-world implementations. Let me know if you need more! 🚀