JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

Major Project Title

Posture Guard : A Posture Detection & Alarming System

Enrolment Number	Name of Student
20103236	Aman Jain

Course Name: Major Project 2

Program: B. Tech. in Computer Science & Engineering

8th Semester

2023-2024

TABLE OF CONTENTS

Date: 1 May 2024

DECLARATION BY THE STUDENT

I hereby declare that the work reported in the Bachelor's Project entitled **"POSTURE GUARD : A POSTURE DETECTION AND ALARMING SYSTEM"** submitted at **Jaypee Institute of Information Technology, Noida, India,** is an authentic record of my work carried out under the supervision of **Dr. Shikha Jain**. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of this project.

Signature :

Name : Aman Jain

Enrolment Number : 20103236

Signature :

Supervisor's Name : Dr. Shikha Jain

Department of Computer Science & Engineering and Information Technology

Jaypee Institute of Information Technology, Noida, India

Date: 1 May 2024

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Bachelor's project entitled "**POSTURE GUARD : A POSTURE DETECTION AND ALARMING SYSTEM**" submitted by **"AMAN JAIN"** at **Jaypee Institute of Information Technology, Noida, India,** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Signature :

Dr. Shikha Jain (Assistant Professor)

Department of Computer Science & Engineering and Information Technology

Jaypee Institute of Information Technology, Noida, India

Date: 1 May 2024

ACKNOWLEDGEMENT

This project was filled with lots of opportunities as well as challenges, and it took an enormous amount of dedication and hard work involved in this project to make it a success. It would not be possible without my friends and family who supported me and were there to have my back on every step of this project. I sincerely extend my heartfelt thanks to all of them. Firstly, I would like to express my humble gratitude to **Dr. Shikha Jain**, for providing me with an opportunity to work under her supervision. Her encouragement, full support, and invaluable suggestions throughout this dissertation were highly valuable. The excellence brought at each step of my work helped me during my project and the writing of this report. Her patience, unwavering courage, and sincerity has consistently served as my source of inspiration.

Signature :

Aman Jain

20103236

ABSTRACT

Post pandemic, i.e, in today's time, prolonged sitting and bad posture have become an increasingly pressing issue, leading to various musculoskeletal issues and disorders. To address this problem, our project introduces the "Posture Guard" system, an innovative solution that takes advantage of computer vision and machine learning techniques to monitor and improve users' posture in real-time. The Posture Guard system uses a webcam to capture live video feed of users, which is then processed using deep learning algorithms to detect human face presence and analyze posture. Through the combination of facial detection and pose estimation, the system classifies posture into predefined categories such as "Good Posture" "Poor Posture" and specific posture-related warnings.

Key features of our system include :

- Real-time posture analysis and feedback to users.
- Audio and visual alerts to notify users of poor posture.
- Minimization of the main application window with automatic pop-up notifications while the app runs in the background.

The system's vision is to promote awareness of good posture habits and encourage users to maintain healthy postures throughout their daily computer usage. By providing instant feedback and reminders, the Posture Guard system motivates the users to adopt better posture habits, ultimately contributing to their improved musculoskeletal health and overall well-being.

LIST OF FIGURES

LIST OF SYMBOLS & ACRONYMS

ACRONYM/SYMBOL	**FULL FORM**
CNN	CONVOLUTIONAL NEURAL NETWORK
DNN	DEEP NEURAL NETWORK
TK	TKINTER
DRHN	DEEP RECURRENT HIERARCHICAL NETWORK
HPO	HYPERPARAMETER OPTIMIZATION
EXE	EXECUTABLE FILE
UI	USER INTERFACE

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL INTRODUCTION

In this era, which is dominated by machines and computers, prolonged screen time and maintaining a good posture have emerged as crucial yet often overlooked aspects of overall health and well-being.[4] Poor posture while using computers is not only contributing to musculoskeletal issues such as back pain and neck strain but are also impacting user's confidence, productivity, and overall quality of life. Recognizing the need of addressing this problem, our project introduces the "Posture Guard" system – a comprehensive solution designed to monitor, analyze, and improve users' posture in real-time.

The Posture Guard system makes use of cutting-edge technologies like computer vision, machine learning, and real-time feedback mechanisms to provide users with actionable insights on their posture habits. By employing the inbuilt webcam, the system captures live video feed of user's sitting postures, which is then processed using sophisticated algorithms to assess posture quality. Through the use of image processing techniques, the system can detect key indicators of good or poor posture, including spinal alignment, head position, and body angles.

Key features of the Posture Guard system include :

- **Real-time Posture Analysis:** Users receive instant feedback on their posture, including visual cues and audio alerts, allowing them to make immediate adjustments for improved posture.
- **Data Tracking and Insights:** Users can track their posture habits over time, view trends, and receive insights into their progress toward achieving better posture.

In addition to its core functionalities, the Posture Guard system also offers a user-friendly interface, customizable settings, and smooth integration with existing digital devices and platforms.

Through the implementation of this Posture Guard system, we want to address the growing need for effective posture management solutions in today's technology-driven world.

1.2 PROBLEM STATEMENT

In today's society, prolonged screen time, and poor ergonomic practices have led to a major problem of inappropriate posture among individuals of all ages. Poor posture, characterized by slouching, forward head position, and spinal misalignment, not only contributes to musculoskeletal discomfort and pain but also poses long-term health risks.[2][3][4]

Despite the growing awareness of the importance of good posture, many individuals lack the knowledge, motivation, and tools to effectively monitor, and improve their posture habits. Existing posture management solutions often suffer from limited accuracy, accessibility, and user engagement, making it very challenging for individuals to adopt sustainable posture-improving behaviors.

The problem statement for this project include the following key aspects:

- **Lack of Awareness and Education:** Most of the people are unaware of the significance of maintaining a good posture and lack access to resources and tools to learn about proper posture habits.
- **Limited Monitoring and Feedback:** Existing posture monitoring solutions often lack real-time feedback and fail to provide actionable insights into users' posture habits, making it difficult for individuals to make meaningful improvements.
- **Ineffective Intervention Strategies:** Traditional methods of posture correction, such as sensor embedded chairs or periodic reminders, may offer short-term benefits but they fail to address the main causes of poor posture.
- **Need for Technological Innovation:** With the presence of digital devices everywhere and advancements made in computer vision and machine learning, there is a pressing need for some innovative posture management solutions that use these technologies to provide personalized and effective posture support.

Addressing these challenges requires the development of a comprehensive posture management system which combines advanced technologies with user-centered design principles to deliver accurate, reliable posture monitoring. By tackling these issues, the project aims to empower individuals to take proactive steps toward improving their posture, enhancing their overall health and well-being.

1.3 SIGNIFICANCE/NOVELTY OF THE PROBLEM

SIGNIFICANCE

The problem of poor posture and the health risk associated with it have significant effects on individuals and society as a whole. Understanding the importance of this problem is essential for recognizing the urgency of developing an effective posture management.

Some key aspects of the problem's significance include:

- **Health Impacts:** Poor posture is a leading cause of musculoskeletal discomfort, pain, and injuries, affecting millions of individuals worldwide. Prolonged periods of slouching or improper sitting can lead to chronic conditions such as back pain, neck pain, and spinal disorders, diminishing an individual's productivity.
- **Long-Term Health Risks:** Beyond immediate discomfort, poor posture contributes to long-term health risks, including spinal misalignment, reduced mobility, and increased proneness to musculoskeletal disorders such as slipped discs. Addressing posture-related issues at an early stage can prevent the development of these health conditions.
- **Impact on Work Performance:** In the workplace, poor posture can impair cognitive function, decrease work efficiency, and increase the risk of injuries. Employees suffering from chronic posture-related pain are more likely to take a sick leave, resulting in productivity losses and economic costs for employers and organizations.
- **Quality of Life:** Good posture is essential for maintaining overall health, vitality, and well-being. Individuals with proper posture experience improved energy levels, mood, and self-confidence which enables them to engage fully in daily activities.
- **Economic Burden:** The economic losses associated with poor posture, including healthcare expenditures and lost productivity represent a significant burden on society. Implementing cost-effective posture management solutions can reduce these losses.

In summary, addressing the problem of poor posture is not only crucial for preventing musculoskeletal disorders and enhancing individuals' quality of life but also for promoting sustainable healthcare practices and economic prosperity. By investing in innovative posture management technologies and initiatives, stakeholders can empower individuals to prioritize their musculoskeletal health, reduce healthcare disparities, and build healthier, more resilient communities.

NOVELTY

The project on posture management and detection has several novel features that distinguish it from existing solutions:

- **Multi-Modal Approach:** Unlike the traditional approaches which focus only on posture, this project uses a multi-modal approach that integrates computer vision, machine learning, and real-time feedback mechanisms. By combining these technologies, the system can accurately detect poor posture and provide alerts or timely interventions to users.
- **Real-Time Feedback:** One of the project's key features is its ability to provide real-time feedback to users about their posture. Using live video feed from embedded webcams, the system continuously monitors users' body positions and delivers instant alerts when inappropriate postures are detected.
- **Integration with Daily Activities:** Rather than just posture correction devices and applications, this project aims to smoothly integrate posture management into user’s daily routines and environments. By embedding posture detection and feedback mechanisms into existing digital platforms and workplace environments, the system promotes continuous awareness and engagement with postural health without disrupting the user's workflow or activities.
- **User-Centric Design:** Throughout the development, the project gives priority to a user-centric design approach, ensuring that the system's interface, features, and interactions are intuitive, accessible, and engaging for diverse user groups. By validating user feedback and conducting usability tests the project aims to create a posture management solution that aligns with users and promotes long-term benefits.

In summary, the project's novelty lies in its overall approach to posture management, real-time feedback, alerts, integration with daily activities and the user-centric design. By pushing the boundaries of traditional posture correction methods and making use of cutting-edge technologies, the project aims to revolutionize how individuals monitor, and improve their postural health in this digital age.

1.4 EMPIRICAL STUDY

- **Effectiveness Study:** The accuracy and reliability of the posture detection algorithm was evaluated by comparing its predictions with manual assessments. Collected the data from a sample of users having different postures, and analyzed the agreement between the algorithm's predictions and ground truth observations.

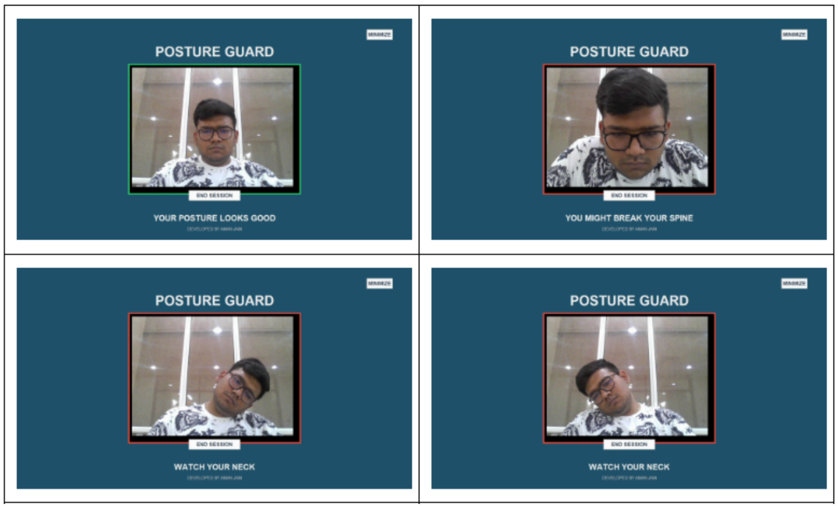

Figure 1. Predictions made by our system.

- **User Experience Study:** Assessed the usability and user experience of the posture detection system through user testing sessions and surveys. Measured factors such as user satisfaction, ease of use, perceived usefulness, and learnability of the system interface.

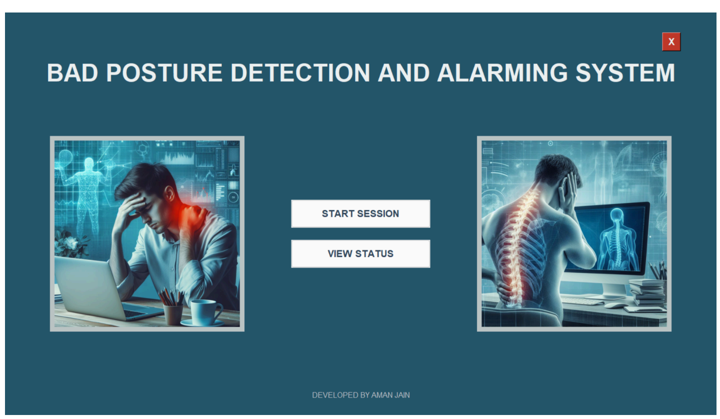

Figure 2. Home page of our system.

- **Impact Study:** Investigated the impact of the posture detection and alarming system on user behavior and posture habits over time. Conducted surveys to evaluate factors such as the user's focus on correcting his posture during daily work and to what extent user's feel the need of having such a posture correcting system.[See Appendix A, page 72]

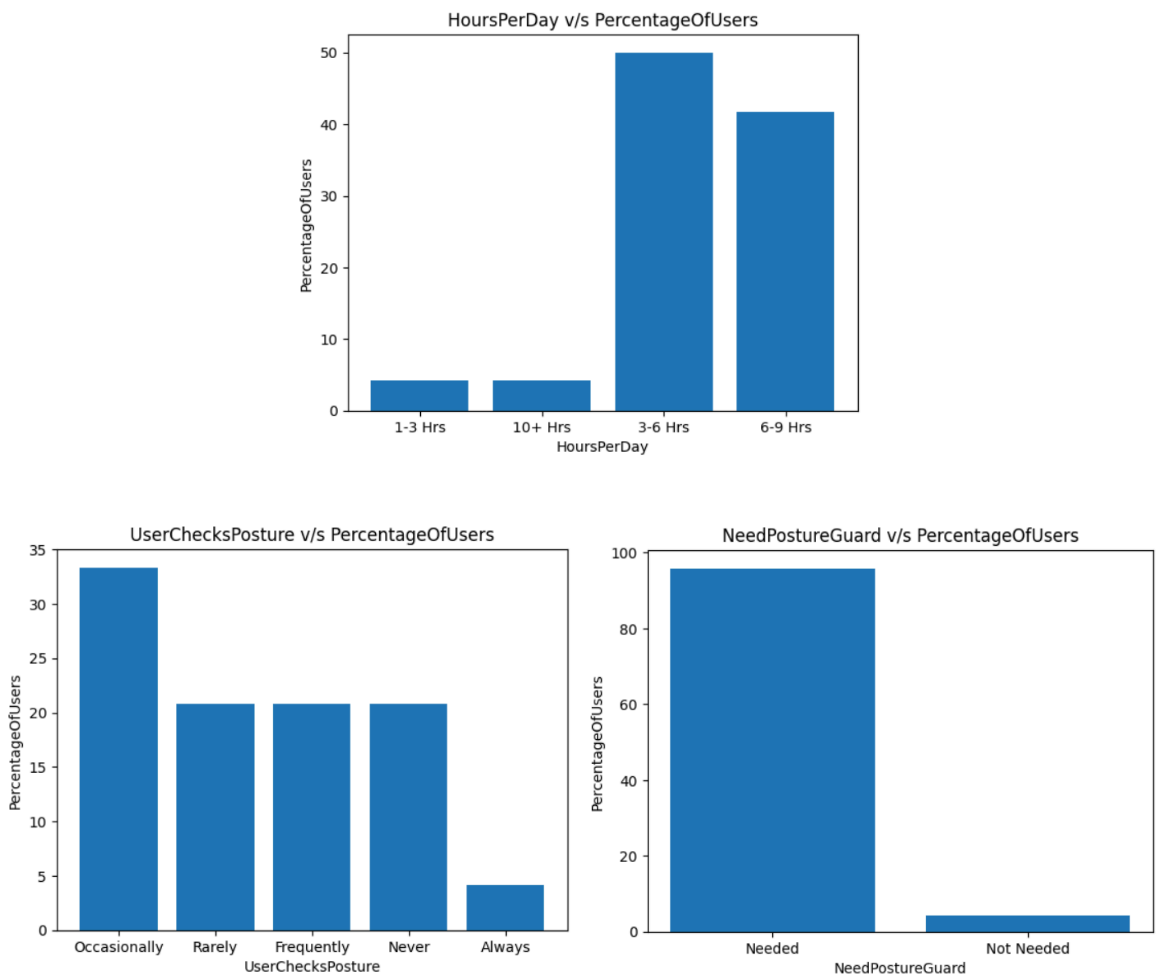

Figure 3. User Survey

- **Comparative Study:** Compared the performance of the posture detection system with existing commercial solutions and alternative approaches. Conducted benchmarking experiments to evaluate factors such as accuracy, speed, resource efficiency, and robustness across different scenarios and users.

1.5 BRIEF DESCRIPTION OF THE SOLUTION APPROACH

The solution approach for this project involves leveraging computer vision and machine learning techniques to develop a posture detection system. The system uses a webcam to capture live video feed, which is then processed using deep learning models to analyze the user's posture in real-time.

Firstly, a convolutional neural network (CNN) model is employed to detect human faces in the video frames. Pre-Trained face detector model from OpenCV library's Deep Neural Network Module (DNN Module) was used for the purpose of detecting human faces in the frame.

Once a face is detected, another CNN model is utilized to classify the posture based on the orientation of key body parts. The system categorizes postures into different classes such as "Correct Posture", "Neck Strain", or "Back Strain" based on predefined criteria.

To enhance user experience, the system provides visual as well as auditory feedback. Visual feedback includes highlighting the detected posture on the screen, while auditory feedback involves playing sound alerts for incorrect postures. Additionally, the system incorporates features such as minimizing the main window during sessions and displaying posture notifications when the window is minimized and running in the background without interrupting the user's work, ensuring user engagement and convenience.

In addition to the solution approach, a significant effort was dedicated to data collection for training the posture classification CNN model. A custom software tool was developed to facilitate the efficient capture of a large dataset comprising around 3000 images distributed across five distinct posture classes. This software streamlines the image capture process, enabling rapid collection within a mere three-minute timeframe. By automating the image acquisition process and categorizing images into predefined classes, the software ensures the availability of a diverse and comprehensive dataset essential for training robust posture classification models. This software was used by 10 people to collect a dataset consisting of almost 40K images divided into 5 classes.

Overall, by combining computer vision, machine learning, and user-friendly interface design, the solution aims to promote awareness of good posture habits and mitigate the risks associated with poor posture-related health issues.

1.6 COMPARISON OF EXISTING APPROACHES

Existing approaches to posture detection and correction vary widely in their methodologies and effectiveness. Traditional methods often rely on manual assessment by healthcare professionals or the use of wearable sensors to monitor posture in real-time. While these approaches can provide valuable insights, they often suffer from limitations such as high cost, inconvenience, and lack of scalability.[3][8]

In recent years, advancements in computer vision and machine learning have paved the way for more automated and accessible solutions. Some existing approaches leverage deep learning techniques to analyze images or video streams in real-time, enabling posture assessment without the need for specialized hardware or manual intervention. These approaches typically involve training convolutional neural networks (CNNs) on large datasets of labeled images to recognize different postures accurately.[2][6]

However, despite their promise, existing approaches still face several challenges. One common issue is the need for large and diverse datasets to train accurate models capable of generalizing across different individuals and environments.[3][4][7][8] Additionally, real-time performance can be a concern, particularly in applications requiring low latency, such as interactive feedback systems or assistive technologies.

Moreover, many existing solutions focus primarily on posture assessment and fail to provide actionable feedback or intervention strategies to help users correct their posture effectively. This limitation underscores the importance of not only accurately detecting poor posture but also guiding users towards adopting healthier habits through feedback mechanisms.

Overall, while existing approaches have made significant strides in posture detection and correction, there remains ample room for improvement in terms of accuracy, real-time performance, and user-centric design. Addressing these challenges could lead to more effective and accessible solutions for promoting better posture and overall musculoskeletal health.

CHAPTER 2
LITERATURE SURVEY

2.1 SUMMARY OF PAPERS STUDIED

[1] Posture Recognition Using Ensemble Deep Models Under Various Home Environments

This study discusses the application of posture recognition using ensemble convolutional neural networks (CNNs) in home environments. With an increasing number of elderly people living alone, posture recognition becomes crucial for ensuring their safety. Traditionally, recognizing posture required obtaining body points' coordinates, depth, and video frame information. However, advancements in deep learning have made it possible to achieve good performance in posture recognition using only an image. The study utilized various pre-trained CNNs, such as VGGNet, ResNet, DenseNet, InceptionResNet, and Xception, and performed experiments based on five types of preprocessing to construct an ensemble deep model combined by majority and average methods. The experiments were conducted using a posture database consisting of 51,000 images with 10 postures from 51 home environments. The results showed that the ensemble system by InceptionResNetV2s with five types of preprocessing exhibited superior performance compared to other combination methods and the pre-trained CNN itself. The construction of a daily domestic posture database was essential, involving 51 homes contributing 100 images per posture, resulting in a total of 51,000 images. Posture recognition was performed using various preprocessed images and CNN models. The CNNs were trained using transfer learning, where most parameters were fixed, and only the final FC layers and Softmax were trained. Moreover, the study explored ensemble methods, including majority vote and score average, to combine the outputs of the deep models and preprocessing types. The results were categorized into standing, sitting, lying, and lying crouched postures, demonstrating the versatility of the approach. The study's significance lies in addressing the need for non-intrusive posture recognition methods, particularly for elderly individuals living alone, and the effectiveness of ensemble deep models in achieving superior performance in posture recognition under various home environments, offering potential applications in security and safety monitoring. In conclusion, the study provides valuable insights into the application of ensemble deep models for posture recognition in home environments. By leveraging pre-trained CNNs and various preprocessing types, the research demonstrates enhanced performance in recognizing different postures, offering potential implications for safety and security monitoring in home environments. Further research in this area could focus on refining the ensemble methods and exploring real-world implementation for assisting elderly individuals living alone.

[2] Human Posture Detection Using Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms

The research paper meticulously examines the intricate challenges associated with employing deep CNN models for human posture recognition. It not only identifies these challenges but also puts forth a novel three-phase model aimed at providing decision support to mitigate these hurdles effectively. The crux of the paper revolves around the pivotal role played by transfer learning, image data augmentation, and hyperparameter optimization (HPO) in enhancing the classification prowess for human posture detection across various renowned models such as AlexNet, VGG16, CNN, and MLP, leveraging the MPII human pose dataset.

One of the standout achievements of the study lies in the remarkable accuracy rates attained: 91.2% with AlexNet, 90.2% with VGG16, 87.5% with CNN, and 89.9% with MLP. These results underscore the efficacy of the proposed methodologies in tackling the inherent complexities of deep CNN models. Moreover, it marks a significant milestone as the first endeavor to execute HPO on the MPII human pose dataset, thus breaking new ground in the realm of posture recognition research.

The paper diligently delves into the myriad challenges posed by deep CNN models, ranging from the pervasive issues of overfitting to the suboptimal performance, particularly evident when deploying expansive models like AlexNet and VGG16 with limited training data. To counter these challenges head-on, the paper advocates for the amalgamation of CNN transfer learning, image data augmentation, and hyperparameter optimization within a cohesive three-phase model. It underscores the transformative potential of HPO methods in shaping the behavior of training algorithms and subsequently influencing the performance metrics of machine learning and deep learning models.

Furthermore, the paper meticulously presents the implementation results of the transfer learning models, both in isolation and in conjunction with image augmentation or HPO. Through meticulous experimentation, it illuminates the nuanced impact of different hyperparameter combinations on model performance, thereby delineating certain combinations with a profound effect while others exert a more marginal influence. Additionally, the paper conducts a comparative analysis with prior studies on posture recognition utilizing the MPII dataset, thereby showcasing the superior efficacy of the proposed approach.

In summation, the study offers an exhaustive analysis of the proposed three-phase model for decision support in human posture recognition. It underscores the indispensable role played by transfer learning, image data augmentation, and hyperparameter optimization in attaining stellar accuracy rates in posture classification. With its comprehensive insights and innovative methodologies, the paper stands as a seminal contribution to the burgeoning field of human posture recognition and deep learning techniques.

[3] Detection of sitting posture using hierarchical image composition and deep learning

The paper proposes a novel deep recurrent hierarchical network (DRHN) model based on MobileNetV2 for human posture detection, specifically focusing on sitting posture recognition. The authors highlight the importance of human posture detection for various applications, including assisted living, healthcare, and physical exercising. They emphasize the limitations of existing approaches, particularly in addressing the occlusion problem when the full human skeleton is not visible in the frame.

The proposed DRHN network is designed to accept RGB-Depth frame sequences and produce a representation of semantically related posture states. The model addresses the occlusion problem related to limited visibility of the human torso by analyzing the temporal features from video frames and using a hierarchical data representation to filter out invalid class labels early in the prediction process.

The authors conducted a pilot study involving 11 test subjects to validate their approach's effectiveness and achieved an impressive 91.47% accuracy for sitting posture recognition at a 10 frames per second

(fps) rate. The study also emphasized the significance of posture recognition in addressing various medical conditions related to sedentary lifestyles, such as back pain and work-related musculoskeletal disorders. Furthermore, the authors developed a dataset for training and evaluation, consisting of 66 different captured video sequence instances. The dataset was used to train the neural network, and the results demonstrated the effectiveness of the proposed approach in real-time applications.
The authors also compared their method with other state-of-the-art posture recognition methods and highlighted the lightweight nature of their approach, making it suitable for real-time applications. The paper addresses the limitations and challenges of the proposed approach, such as the small number of subjects in the pilot study and the need for diverse subject groups in future research. Additionally, the authors discuss the potential for further improvements in dataset collection methodology to account for different body shapes and disabilities and reduce labeling ambiguities. They also provide insights into the computational aspects of their approach, including training optimization methods and data augmentation techniques.

Overall, the proposed DRHN model based on MobileNetV2 demonstrates promising results for real-time sitting posture recognition, addressing the occlusion problem related to limited visibility of the human torso in the frame and achieving high accuracy in posture classification. The authors' thorough methodology and validation procedures contribute to the advancement of human posture detection and offer potential for real-world applications in assisted living and healthcare.

[4] A Deep-Learning Based Posture Detection System for Preventing Telework-Related Musculoskeletal Disorders

The research paper focuses on the impact of teleworking on workers' posture and the development of an automated system to detect and quantify incorrect postural habits while providing recommendations to prevent health problems. The study addresses the increased prevalence of teleworking due to the pandemic and its potential impact on workers' posture and musculoskeletal health. The authors designed, implemented, and tested a system based on the postural detection of the worker using a specialized hardware system that processes video in real-time through convolutional neural networks.

The hardware system used for real-time video processing was capable of detecting the posture of the neck, shoulders, and arms, providing recommendations to prevent potential health problems due to poor posture. The study revealed that the video processing could be carried out in real-time with a low power consumption (less than 10 watts), and obtained an accuracy of over 80% in terms of the pattern detected. The results also indicated that teleworking was associated with an increased risk of musculoskeletal disorders due to incorrect postures in front of the computer.

The paper provided a detailed explanation of the posture and postural hygiene, and the importance of maintaining a correct posture to avoid spinal injuries such as hyperlordosis, hyperkyphosis, scoliosis, or rectifications. The study also discussed the factors affecting posture and the need for automated tools to quantify the degree of incorrectness of a postural habit in a worker. The research findings highlighted the significance of developing recommendation systems and tools for the early detection of bad posture to prevent injury and promote postural hygiene.

Additionally, the study evaluated the performance of the system using multiple hardware platforms and testing different scenarios including real-time video processing and offline video classification. The results showed that the system was capable of recognizing correct posture with a high level of accuracy, and the efficiency and suitability of different hardware platforms for the system were also assessed.

Overall, the research paper provided valuable insights into the impact of teleworking on workers' posture and the development of an automated posture detection system. The study's findings offer a potential solution to address the challenges related to postural health in the context of teleworking, and the evaluation of different hardware platforms provides practical recommendations for implementing the system in real-world scenarios.

[5] Modeling Proper & Improper Sitting Posture of Computer Users Using Machine Vision for a Human-Computer Intelligent Interactive System During Covid-19

The paper discusses a study on modeling proper and improper sitting posture of computer users using machine vision for a human-computer intelligent interactive system during COVID-19. The study aimed to develop a system for sitting posture monitoring in a work-from-home setup and for rehabilitation purposes, addressing the impact of the COVID-19 pandemic on traditional working environments. Human posture recognition, a challenging task in computer vision, was achieved using a small-scale convolutional neural network and a smartphone built-in camera, with the study demonstrating accuracies of 85.18% and 92.07%, and kappas of 0.691 and 0.838, respectively. The study also discussed the impact of sedentary behavior, health risks associated with prolonged sitting, and the need for monitoring proper sitting posture, especially in work-from-home setups. The study's methodology involved creating a custom dataset for sitting posture recognition, developing a model using a rule-based approach, and evaluating the impact of ergonomic and demographic elements on sitting posture. Data gathering was conducted using two mobile phones and a web camera to monitor participants' sitting posture, and the resulting videos were annotated by licensed physical therapists. The data was then processed to extract features such as the distance and angles between key points, including the upper and lower back, shoulders, and elbows. The feature extraction tool utilized MediaPipe for high-fidelity body pose tracking and achieved superior performance in mean average precision and Percentage of Correct Key-points compared to existing solutions. The study's findings are significant for addressing the challenges posed by the work-from-home setup during the COVID-19 pandemic, providing insights into the impact of sedentary behavior on health and the potential benefits of posture recognition systems in mitigating these negative effects. The development of a system capable of monitoring proper and improper sitting posture has promising implications for improving overall well-being and productivity in work-from-home environments.

[6] Computer Users Sitting Posture Classification Using Distinct Feature Points and Small Scale Convolutional Neural Network for Humana Computer Intelligent Interactive System During COVID-19

The research paper discusses the development of an intelligent and interactive system for classifying proper and improper sitting postures using distinct feature points and small-scale convolutional neural networks. It addresses the impact of the COVID-19 pandemic on work practices, particularly the transition to work-from-home setups, and the resulting effects on overall health. The study aims to develop a model that utilizes human pose estimation to assess proper and improper sitting postures in a work-from-home environment. It utilizes distinct key points such as thoracic, thoraco-lumbar, and lumbar points in the spine to recognize body key points. The study developed and implemented a small-scale convolutional network and low-cost smartphone camera to recognize these key points and utilized additional features such as cosine similarity and point distances to classify proper and improper sitting postures. The study developed (2) datasets and (2) models with an accuracy of 85.18% and

92.07% and a kappa of 0.691 and 0.838, respectively. The study highlighted the challenges and opportunities in human pose estimation and the importance of assessing sitting postures for employees in a work-from-home setup. The paper discusses the challenges and advancements in human pose estimation using deep learning and specialized cameras. It outlines the progress and applications of human pose estimation in various fields such as animation, human monitoring, video surveillance, and assistance systems for daily living and driver systems. The document also discusses the use of various sensors and vision-based methods to assess risk factors for musculoskeletal disorders (MSDs) in relation to posture. It explores the limitations of current sensing and monitoring methods and emphasizes the need for non-intrusive devices and less expensive solutions. Additionally, it presents several studies that have utilized pressure sensors, accelerometer sensors, and vision-based techniques to assess sitting postures. The paper provides insights into the challenges and opportunities in developing posture recognition systems using virtual markers and cameras, and the importance of considering the entire body in the classification of sitting postures. The document outlines the materials and methods used in the study, including data gathering, keypoint extraction, data preparation, and pose estimation results. It discusses the recognition of raw feature points and the performance of the recognition model. The study evaluated the quality of the recognition model using mean average precision (mAP) and Percentage of Correct Keypoints (PCK). The results showed that the model outperformed other existing solutions in terms of mAP and PCK. The paper also presents the decision tree rules for the left and right camera models, highlighting the significant attributes and ergonomic elements used in the classification of proper and improper sitting postures. Overall, the paper provides a comprehensive overview of the development of an intelligent and interactive system for classifying proper and improper sitting postures using distinct feature points and small-scale convolutional neural networks. It highlights the significance of human pose estimation in assessing sitting postures, particularly in work-from-home setups, and the potential impact on overall health and productivity. The study contributes to the advancement of technology in addressing ergonomic and health-related challenges in remote work environments during the COVID-19 pandemic.

[7] Recurrent Network Solutions for Human Posture Recognition Based on Kinect Skeletal Data

The paper discusses a study on the use of Recurrent Neural Networks (RNNs) for human posture recognition based on Kinect skeletal data, particularly in the context of Ambient Assisted Living (AAL) systems. The introduction outlines the growing need for monitoring frail individuals, especially the elderly, and the potential of AAL systems in addressing this demand. The document highlights the advantages of using low-cost RGB-D devices like Kinect V2, which extract skeletal data for monitoring human activities in a less intrusive manner. Moreover, it emphasizes the role of deep learning-based algorithms, particularly RNNs, for automatically identifying human postures in home monitoring systems. The study explores the performance of different RNN models, including 2BLSTM and 3BGRU, in classifying time series of skeletal tracking data to identify daily living postures and potentially dangerous situations. Additionally, the document discusses the feature sets used for training the models, such as human-crafted kinematic features and ego-centric 3D coordinates of skeletal joints, along with the application of data augmentation methods to improve the generalization ability of the models. The study presents the results obtained from the different RNN models applied to various datasets. For the first dataset with eight features and five classes, the 3BGRU architecture achieved a mean accuracy of 82% and demonstrated high specificity values for all classes, indicating a low rate of false positives. However, there were variations in sensitivity and precision across different classes, suggesting differences in the model's ability to identify true positives and produce false positives for each class. In the second dataset with 52 features and five classes, the 3BGRU architecture achieved a mean accuracy of 81% with high specificity values. Notably, the model faced challenges in correctly identifying true positives for Class 5, resulting in low sensitivity values. Furthermore, the mean precision values indicated potential issues in distinguishing between true positives and false positives, particularly for Class 5. The third dataset, also with 52 features but four classes, saw the 3BGRU architecture achieving a mean accuracy of 87%. The model demonstrated high specificity and sensitivity values for most classes, with relatively high precision. However, Class 4 had lower sensitivity and precision values, indicating challenges in correctly identifying and distinguishing true positives and false positives for this class. Overall, the study provides insights into the performance of different RNN models and feature sets for human posture recognition based on Kinect skeletal data, shedding light on the potential and challenges of using deep learning algorithms in AAL systems. It also highlights the importance of addressing class-specific performance variations and precision issues for practical applications in home monitoring systems.

[8] A Method For Complex Posture Recognition During Long-Term Sitting Using Neural Networks And Pressure Mapping Systems

The document discusses a study on the identification of correct sitting postures during prolonged periods using a pressure mapping system and Convolutional Neural Networks (CNN). The study was conducted in three stages: data collection using two pressure mapping systems, training a CNN model to represent each posture, and long-term monitoring with feedback to participants. The results showed that the system accurately identified the three postures with an accuracy of 0.854 and provided understandable and helpful feedback to improve posture. The study demonstrated the potential of the approach to facilitate further research on sitting posture, as it can be easily adapted to various pressure mapping systems without altering the methodology. The study aimed to address the common occurrence of poor sitting posture and its potential impact on discomfort and pain, particularly among office workers with sedentary lifestyles. Various methods, including pressure sensor arrays and individual sensors, have been used to detect sitting postures. The study also highlighted the importance of understanding sitting postures over time to promote correct posture, prevent injuries, and provide tools for researchers and physicians interested in preventing long-term injuries. Furthermore, the methodology used in this study has the potential to facilitate additional research on sitting posture due to its versatility and adaptability to various pressure mapping systems. The study presented a reliable approach to detecting and categorizing sitting postures, which can reduce pain and discomfort, especially among individuals who remain seated for extended periods. Additionally, the study demonstrated the feasibility and versatility of the algorithms to be used with different pressure mapping systems through the implementation of Transfer Learning. The feedback from participants also indicated the system's acceptance and usability, suggesting its potential for improving overall well-being. Overall, the study provides valuable insights into the detection and categorization of sitting postures, offering a robust methodology that can be easily adapted and has the potential to promote correct posture and prevent discomfort and injuries associated with prolonged sitting.

2.2 INTEGRATED SUMMARY OF THE LITERATURE

PAPER	DESCRIPTION	PROS	CONS
[1]	The study discusses the use of ensemble deep models for posture recognition in various home environments, focusing on the application of convolutional neural networks and the potential benefits for elderly people living alone.	One pro of this study is that the experimental results reveal that the ensemble deep model shows good performance in comparison with the pre-trained CNN itself.	limitations of input methods due to the neural networks being restricted to a square fixed image, which can result in tightness or distortion of the person in the images.
[2].	The document discusses a hyperparameter optimization methodology for decision-making in the medical field using deep learning approaches, specifically Convolutional Neural Networks (CNNs), to improve human posture image classification and support doctors in making critical clinical decisions more effectively.	The study presents a new decision support framework for the optimization of hyperparameters for various models, which can lead to optimal classification results and improve the effectiveness of critical clinical decisions.	The increased complexity of finding optimal hyperparameter values requires additional computational resources, which can be a disadvantage of the study.
[3]	The main idea of the document is the development of a novel deep recurrent hierarchical neural network approach for tracking human posture in home office environments, with a focus on classifying and improving prediction of forward and backward postures using a small dataset of 11 test subjects.	One main advantage of this study is the use of a novel deep recurrent hierarchical neural network approach for tracking human posture in home office environments, which has the potential to improve the accuracy of posture classification.	One main limitation of this study is the small number of subjects (11) and the lack of diversity in the subject group, which may have influenced the validity of the results. Additionally, the dataset is slightly skewed towards certain postures, which may impact the generalization of the network.

PAPER	DESCRIPTION	PROS	CONS
[4]	The document discusses the development and validation of a system that uses a classical GPU and specialized hardware to autonomously recognize and provide real-time information on joints' angles and positions, with potential for further study and application.	One main pro of this study is that the results show a posture detection accuracy of over 80% for the 4-class original problem and more than 90% for the 2-class classification system.	One main con of this study is that the evaluation was conducted in a laboratory environment with participants belonging to the research group that developed the work. This may limit the generalizability of the results to a broader population and real-world settings.
[5]	The paper discusses the process of gathering and analyzing data on computer users' sitting posture using key points and cameras, with a focus on identifying proper and improper posture to inform ergonomic recommendations.	The study successfully recognized proper and improper sitting postures using virtual markers and a small-scale convolutional neural network.	The investigation on the use of less expensive devices, less intrusive, and less computationally expensive methods is needed, suggesting potential limitations in the current setup and equipment used for data gathering.
[6]	The main idea of the study is the development and implementation of a small-scale convolutional network and low-cost smartphone camera to recognize body key points for assessing proper and improper sitting posture in a work from home environment.	The main pro of the paper is the development of an intelligent and interactive system utilizing a human estimation model with the use of distinct feature points and a small scale convolutional neural network.	The main con of the paper is the potential limitation of the small scale convolutional neural network, as it may not be as powerful as high performing machines.

PAPER	DESCRIPTION	PROS	CONS
[7]	The document discusses the development and testing of various deep learning models for classifying human activities based on data collected from different subjects and the potential for further improvement in performance through the use of larger training datasets and alternative deep learning models.	The study explores the use of different feature selection methods and machine learning architectures for the classification block. Additionally, it evaluates the performance of different models in correctly identifying specific sequences of postures.	The study had a small number of sequences used for model training, which may limit the generalization ability of the model. Collecting more data could improve the model's ability to capture complex patterns.
[8]	The main idea of the document is to use a pressure mapping system and Convolutional Neural Networks (CNN) model to identify and improve correct sitting postures during long periods of sitting, with a focus on posture recognition, feedback, and overall well-being.	The main pro of this study is that it provides a systematic approach to identifying correct sitting postures using a pressure mapping system and Convolutional Neural Networks (CNN).	One potential con of the study is that it was conducted on a relatively small sample size, with only 22 volunteers participating in the first stage. This limited sample size may impact the generalizability of the findings to a larger population.

Table 1. Integrated Literature

CHAPTER 3
REQUIREMENT ANALYSIS AND SOLUTION

3.1 OVERALL DESCRIPTION OF THE PROJECT

The project focuses on developing an advanced posture detection and monitoring system aimed at addressing the growing concerns surrounding poor posture and its adverse effects on health, particularly in the context of increased teleworking and sedentary lifestyles. Leveraging state-of-the-art technologies such as convolutional neural networks (CNNs), deep learning models, and real-time video processing, the system aims to provide real-time feedback and recommendations to users to improve their posture habits and mitigate the risk of musculoskeletal disorders.

The project encompasses several key components, including the implementation of sophisticated machine learning algorithms for posture detection and classification, and the design of user-friendly interfaces for seamless interaction and feedback delivery. Through meticulous data collection, model training, and validation procedures, the system strives to achieve high accuracy and reliability in posture recognition while ensuring minimal computational overhead and power consumption.

Furthermore, the project emphasizes the importance of user education and awareness regarding proper posture habits, incorporating features such as progress tracking to empower users to take proactive steps towards improving their posture and overall well-being. Additionally, the system's scalability and adaptability enable its deployment across various settings, including home offices, workplaces, and healthcare facilities, thereby catering to a diverse range of users and use cases.

Overall, the project represents a comprehensive effort to tackle the challenges posed by poor posture and its associated health risks in modern society. By integrating cutting-edge technologies with user-centric design principles, the system aims to promote postural hygiene, reduce the incidence of musculoskeletal disorders, and enhance the overall quality of life for users across different demographics and environments.

3.2 REQUIREMENT ANALYSIS

Requirement Analysis for Posture Detection and Monitoring System

Functional Requirements:

- Real-time Video Processing: The system should be capable of capturing live video feeds from cameras in real-time.
- Posture Detection: Implement algorithms for detecting and classifying various postures, including correct and incorrect postures.
- Feedback Delivery: Provide real-time feedback to users regarding their posture status and recommendations for improvement.
- User Interface: Develop user-friendly interfaces for interacting with the system, including desktop applications.
- Notification System: Implement a notification system for alerting users of posture-related issues and providing timely reminders.
- Integration with Hardware: Ensure compatibility and integration with specialized hardware systems for video capture and processing.

Non-Functional Requirements:

- Accuracy: Achieve high accuracy in posture detection and classification to ensure reliable feedback and recommendations.
- Performance: Ensure the system's performance meets real-time processing requirements with minimal latency.
- Scalability: Design the system to scale effectively to accommodate varying numbers of users and data volumes.
- Security: Implement robust security measures to protect user data and ensure privacy compliance.
- Usability: Prioritize user experience by designing intuitive interfaces and providing clear instructions for interaction.
- Reliability: Ensure system reliability and stability to minimize downtime and ensure continuous operation.

- Compatibility: Ensure compatibility with a wide range of devices, operating systems, and browsers to maximize accessibility.

Hardware Requirements:

- Cameras: Compatible cameras for capturing live video feeds with sufficient resolution and frame rate.
- Processing Units: High-performance processors and graphics processing units (GPUs) for real-time video processing.
- Memory: Adequate memory capacity to handle data processing and storage requirements.

Software Requirements:

- Development Tools: Programming languages, libraries, and frameworks for developing algorithms, user interfaces, and backend systems.
- Machine Learning Frameworks: Libraries and frameworks for implementing machine learning algorithms for posture detection and classification.

Environmental Requirements:

- Physical Environment: Consider environmental factors such as lighting conditions and camera placement for optimal posture detection.

3.3 SOLUTION APPROACH

Solution Approach for Posture Detection and Monitoring System

- **Data Collection and Preparation:** Gather a diverse dataset of images or video clips depicting various postures, including correct and incorrect ones. Annotate the dataset with labels indicating the posture category for supervised learning.
- **Algorithm Development:** Develop deep learning algorithms, such as convolutional neural networks (CNNs), for posture detection and classification. Train the algorithms using the annotated dataset to learn the features associated with different postures. Experiment with different architectures, hyperparameters, and optimization techniques to improve model performance.
- **Real-time Video Processing:** Implement a system for capturing live video feeds from cameras in real-time. Preprocess the video frames to enhance quality, reduce noise, and standardize format for input to the posture detection algorithms. Apply the trained posture detection models to analyze the video frames and classify the observed postures.
- **Feedback Delivery:** Develop mechanisms for providing real-time feedback to users based on the detected postures. Design user interfaces, such as desktop applications or mobile apps, to display posture status, recommendations, and alerts.
- **Notification System:** Implement a notification system to alert users of posture-related issues and provide timely reminders. Define criteria for triggering notifications based on posture analysis results, such as prolonged incorrect posture or lack of movement. Deliver notifications through various channels, including pop-up messages, sound alerts, or vibration alerts on mobile devices.
- **User Interaction and Customization:** Enable user interaction with the system through intuitive interfaces and controls. Allow users to customize feedback settings, notification preferences, and other system parameters according to their needs. Provide options for tracking posture history and monitoring progress over time.
- **Testing and Evaluation:** Conduct extensive testing and evaluation of the system's performance under various conditions, including different postures, lighting environments, and user scenarios. Measure accuracy, precision, recall, and other relevant metrics to assess the

effectiveness of posture detection algorithms. Collect user feedback and incorporate suggestions for improvement to enhance usability and user satisfaction.

- **Deployment and Maintenance:** Make the system ready for deployment in real-world environments, such as offices, homes, or healthcare facilities. Monitor system performance and user feedback to identify areas for optimization and enhancement.

CHAPTER 4

MODELING AND IMPLEMENTATION DETAILS

4.1 DESIGN DIAGRAMS

4.1.1 Flowchart

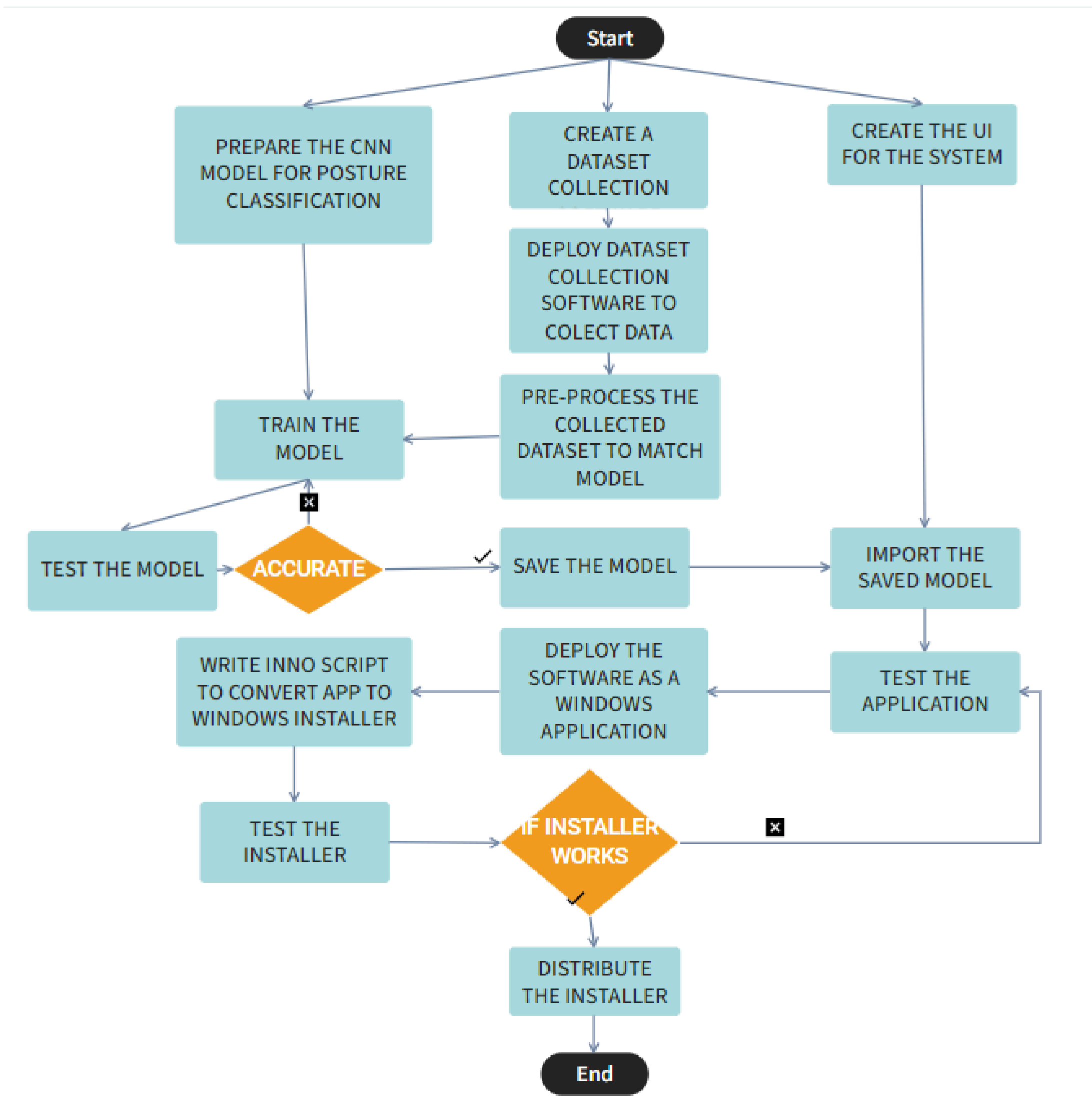

Figure 4. System Flowchart

4.1.2 Sequence Diagram

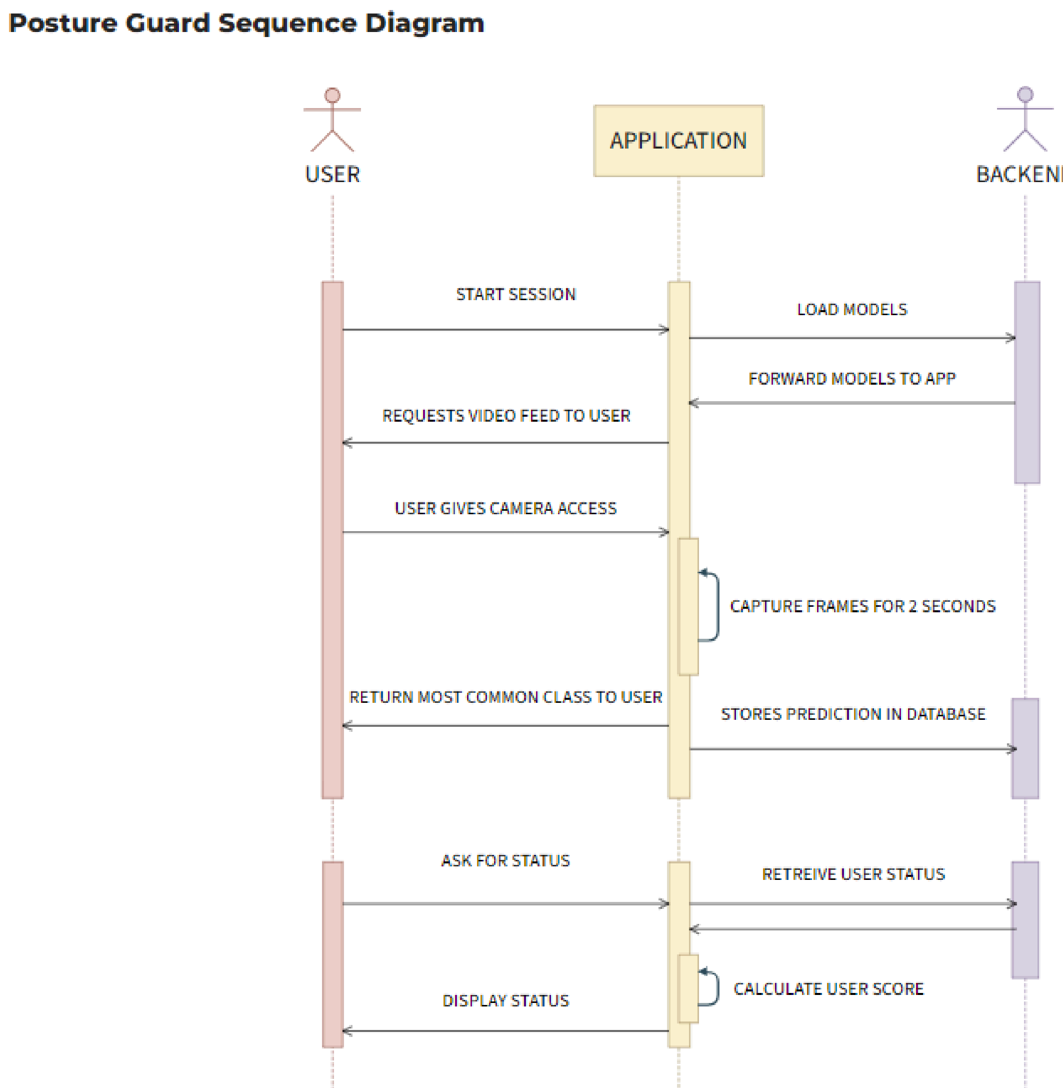

Figure 5. System Sequence Diagram

4.1.3 User Interface Mockup Diagrams

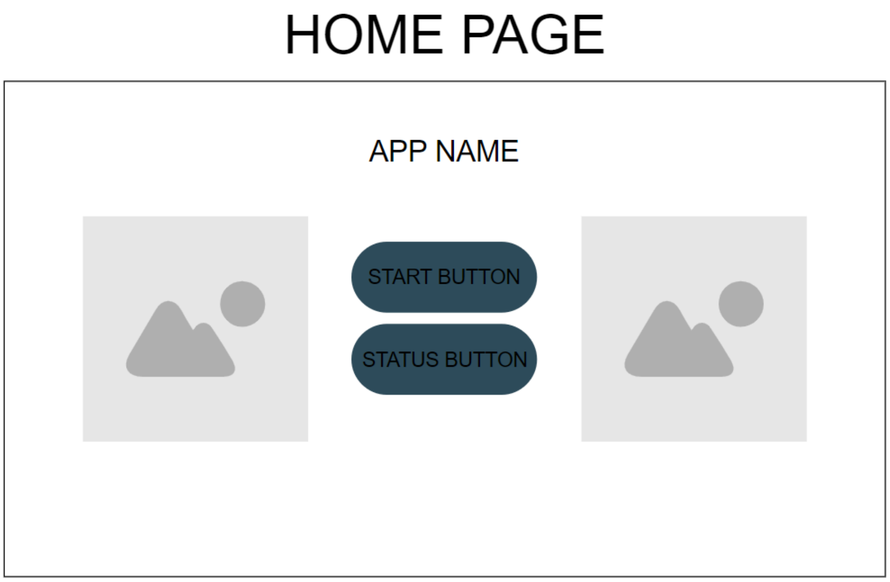

Figure 6. Home Page UI Mockup

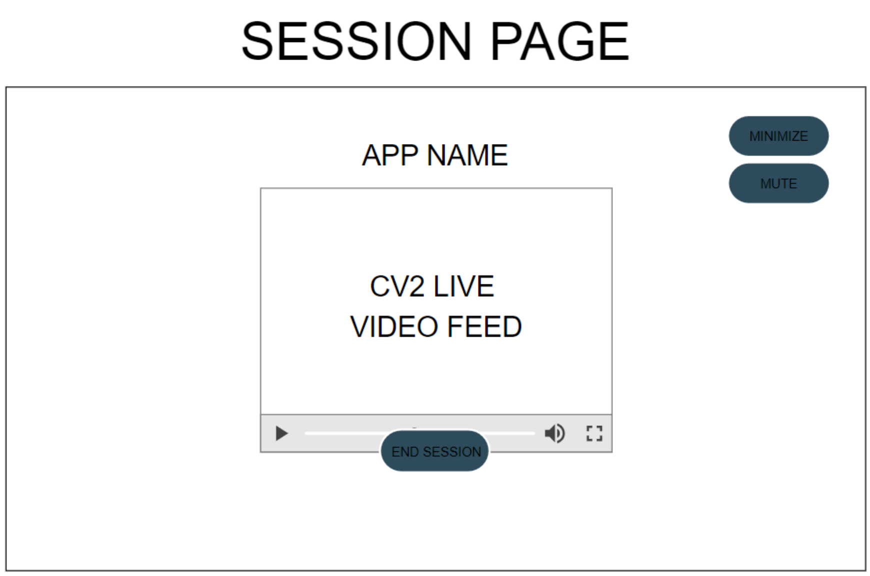

Figure 7. Session Page UI Mockup

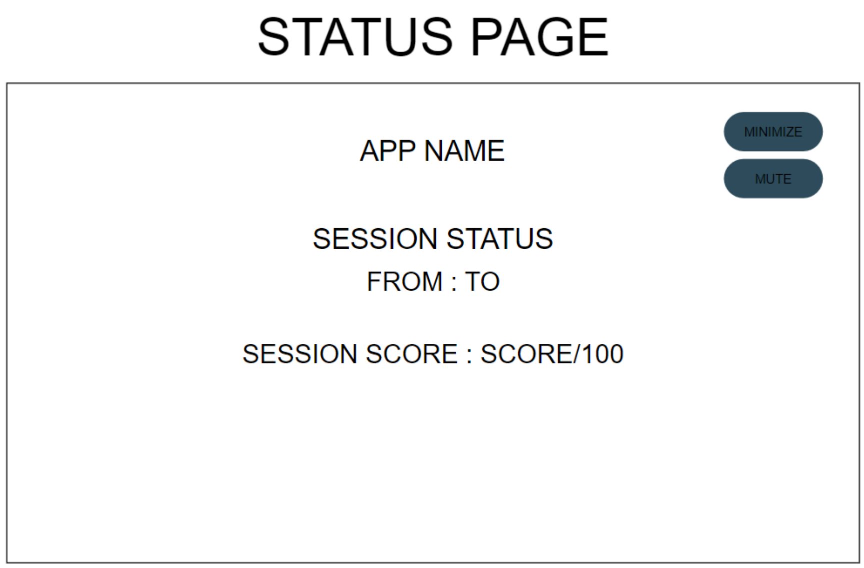

Figure 8. Status Page UI Mockup

4.2 IMPLEMENTATION DETAILS AND ISSUES

4.2.1 INTRODUCTION

The implementation phase of our posture detection and monitoring project marks a crucial stage in translating the conceptual framework into a functional system capable of real-time posture analysis and feedback. This phase encompasses the development and integration of various hardware and software components, leveraging advanced technologies in computer vision, deep learning, and human-computer interaction. The primary objective of this phase is to design, build, and test a robust system capable of accurately detecting and analyzing human posture, providing timely feedback to users, and promoting healthy ergonomic practices.

Throughout this implementation journey, we employed a diverse array of technologies, including convolutional neural networks (CNNs), image processing libraries, hardware devices, and user interface frameworks. These technologies were carefully selected to address the multifaceted challenges inherent in posture detection, ranging from occlusion issues to real-time processing constraints. By leveraging the power of machine learning and computer vision algorithms, we aimed to develop a system capable of identifying various postural abnormalities and providing actionable insights to users for corrective action.

In this section, we provide a comprehensive overview of the implementation process, detailing the key components, methodologies, and challenges encountered along the way. We discuss the data acquisition and preprocessing pipeline, the design and training of posture detection algorithms, the development of the user interface for interaction, and the integration and testing of the system components. Additionally, we reflect on the lessons learned, highlighting successes, failures, and areas for future improvement. Overall, the implementation phase represents a critical milestone in the realization of our posture detection and monitoring system, bringing us closer to our overarching goal of promoting postural health and well-being in both professional and personal settings. Through innovative technologies and rigorous development methodologies, we strive to deliver a solution that empowers users to maintain optimal posture and mitigate the risk of musculoskeletal disorders.

4.2.2 TECHNOLOGIES USED

- **Convolutional Neural Network (CNN)**

Convolutional Neural Networks are the deep neural networks, specifically designed to work on images tasks such as image classification, segmentation, etc. Since the introduction of CNN's, it has been observed that CNN works better on image tasks than any other deep neural networks.

CNN models beat any other deep learning model for image tasks due to several reasons. CNN uses a hierarchical feature learning in which image features are automatically learned by the model during the training process and they need not be specified explicitly, which tremendously decreases the amount of work required. CNN also works faster due to their weight sharing technology. CNN's use locality based weight sharing on images which allows the model to share weights across different parts of the image. CNN's have another advantage of translation invariance, which allows them to learn features and patterns irrespective of their positions in the image.

Overall, the unique architectural design and inherent properties of CNNs make them highly effective and efficient for a wide range of image-related tasks, including image classification, object detection, segmentation, and more.

In our project, Convolutional Neural Networks play the role of backbone as they are used to classify the postures and detect them if the postures seem abnormal.

- **Computer Vision Library (OpenCV)**

OpenCV is a python’s computer vision library which provides a comprehensive set of functions for image processing tasks such as filtering, thresholding, edge detection, and morphological operations. These functions help us to manipulate and enhance images for various applications. OpenCV also supports video input/output operations, allowing it to read, write, and process video streams in real-time. It includes functions for video capture and frame extraction, which is a crucial task in our project. OpenCV also includes implementations of popular object detection algorithms such as Haar cascades, DNN’s and HOG (Histogram of Oriented Gradients) which we have also used for the face detection task before posture classification.

OpenCV's remarkable benefit is that it integrates with machine learning frameworks such as TensorFlow with great ease, allowing us to train and deploy custom machine learning models for various computer vision tasks. It supports popular deep learning frameworks such as TensorFlow, keras, and Caffe. We have used a Caffe model for facial detection and our posture classification model is trained on the TensorFlow framework. Another main reason for using OpenCV is that it is designed to be platform-independent and supports various operating systems including Windows, Linux, macOS, Android, and iOS.

In our project, we have extensively exploited the OpenCV library. Starting from the first step, i.e, dataset collection to the final but important step of detecting postures in real time, OpenCV was deployed. OpenCV was used at every step of this project, like, dataset collection, image pre-processing, facial detection, model testing and finally in the real time detection task.

- **Machine Learning Framework (TensorFlow)**

Tensorflow is an open-source machine learning framework which is developed by Google Brain. It serves as the most widely used framework for building and training machine learning and especially deep learning models.

Tensorflow has several such advantages which makes it an ideal choice for our use. It has a flexible architecture which allows us to train and deploy across different devices including CPU's, GPU's and also TPU's. It uses different graph execution techniques which makes it very fast and ideal for both training and deployment in a production environment.

Overall, TensorFlow is a powerful and versatile framework that enables us to build, train, and deploy machine learning models for a wide range of applications, from computer vision and natural language processing. Its rich set of features, performance, and scalability make it a preferred choice for our Posture Guard system.

- **High Level Neural Network API (Keras)**

Keras is a high level neural networks API written in python and which is capable of running on top of the TensorFlow framework. Keras provides an user-friendly interface which makes it easier to experiment and deploy machine learning models with minimal code and complexity.

Keras offers a modular approach to quickly define and configure deep learning models. Keras provides us with building blocks for layers, activation, and specifying training parameters. Keras has support for a very wide range of neural networks including CNN, RNN, LSTM, etc.

Overall, Keras is a powerful and user-friendly tool for building deep learning models. Its simplicity, flexibility, and integration with popular deep learning frameworks make it a preferred choice for our project implementation.

- **User Interface Framework (Tkinter)**

Tkinter is the Graphical User Interface (GUI) library provided by Python. Tkinter provides a large set of Python bindings for the Tk GUI toolkit, which is a cross-platform toolkit widely used for creating desktop applications with graphical user interfaces.

Tkinter library has several advantages which makes it the ideal choice for use in this project. Tkinter is a cross platform library and is included with most of the python installations which makes it easily available on most of the operating systems like Windows, Linux, Mac etc. Tkinter also provides us with a large set of widgets, which helps us make better user interfaces. These widgets include buttons, labels, canvas, etc. Tkinter makes it the ideal choice because of its event driven programming style which is best suited for creating applications as it allows us to define how our application should respond to human interaction.

Overall, Tkinter provides a versatile and user-friendly toolkit for building GUI applications in Python. Whether you're developing simple utilities, data visualization tools, or complex desktop applications, Tkinter offers a robust foundation for creating visually appealing and interactive user interfaces.

- **PyInstaller and INNO Setup Scripts**

PyInstaller is a very popular Python library which is used for converting Python scripts into standalone executable files(.EXE), which can be run on systems that do not have Python installed. It analyzes the Python script and its dependencies(data like images, ML models, etc), including modules and packages, and bundles them together into a single executable file. PyInstaller supports various operating systems, including Windows, macOS, and Linux, making it a versatile solution for distributing Python applications across different platforms.

INNO Setup is a free and open-source script-driven installation system for Windows programs.
Its scripting language is derived from Pascal language and it provides a flexible and customizable framework for creating professional installation packages for Windows

applications. INNO Setup allows developers to define installation tasks, such as creating directories, creating shortcuts and copying files through a simple script language. It also supports features like compression, and uninstallation support, making it suitable for both simple and complex installation scenarios. Its basic use is to convert our executable(.exe) application into an executable(.exe) installer which is generally smaller in size and makes it easier to distribute the application.

When combined, PyInstaller and INNO Setup offer a comprehensive solution for packaging and distributing Python applications on the Windows platform.
Here's how they are used together in our project :

- PyInstaller: First we used the PyInstaller to convert our Python script(.py app file) into a standalone executable file. PyInstaller analyzes the python script and its dependencies, including all external libraries, modules and data used, and bundles them together into a single executable file. This file has the capability to be distributed and run on any Windows system without the need for Python to be installed but it is quite large in size(approx 1.5GB) which makes it difficult to distribute easily.
- INNO Setup: To conquer this issue, we used INNO Setup to create an installer package for our Python application created by PyInstaller. INNO Setup allowed us to define the installation tasks and customize the appearance of the installer through a script file. We specified the files to be included in the installation package, create shortcuts and configure other installation settings. Once the script was ready, INNO Setup compiled it into a setup executable that can be distributed to users for installation very easily given its smaller size(approx 250MB)

In conclusion, using PyInstaller combined with INNO Setup allowed us to create an executable file for our system that can run on any windows system without the need of explicitly installing python support. It also allowed us to perform this task while keeping the size of file to be distributed very less as compared to standalone executable.

4.2.3 DATA ACQUISITION & PROCESSING

Introduction

In this phase of implementation, we acquired the required dataset for our posture classification model, labeled it for implementing a supervised learning algorithm, pre-processed it so as to train a model on this data. Novelty of our problem statement led to unavailability of reliable dataset for our use. Very limited work has been done in this domain and the dataset used for those models was dumped rather than making them public because of privacy issues. We had to create our very own dataset collection software for acquiring and labeling the required data for model training. This collected dataset was analyzed, filtered and pre-processed into our final dataset.

This was a very crucial phase in implementation of this project as the quality of the dataset will define how well our model would work in the real world scenarios. Several challenges were faced during this phase of implementation which are discussed below.

Data Collection

There were no reliable datasets available in the public domain for training the model. Novelty of the problem and privacy concerns were two main factors which led to this problem. Very less work being done in this field impacted directly to the less availability of dataset. We had to create our own python script which was capable of capturing user images and putting them into five classes on its own.

Here's how the script worked :

- As soon as the script was run, the user was prompted to do 5 postures(Normal, Neck Bend, Forward Bend, Backward Bend and Misaligned Shoulders) at time intervals of 30 seconds.
- These 30 seconds clips were recorded at 20 FPS and the clips were saved in the format UserName_Posture.mp4.
- This led to 5 videos per user who participated in the dataset collection software.
    - User_Normal.mp4
    - User_NeckBend.mp4
    - User_ForwardBend.mp4
    - User_BackwardBend.mp4
    - User_MisalignedShoulders.mp4

- Now, these five videos were fed to OpenCV library which extracted frames from each video and stored them in separate folders named in format Posture/UserName_Posture/images.jpeg
- The 30 seconds clips recorded at 20 FPS led to 600 frames/posture-user, which means nearly 3000 images were captured into five classes for each user who participated in the data collection process.
- The script was run by 10 people which gave us a huge dataset to start with(approx 40K images)

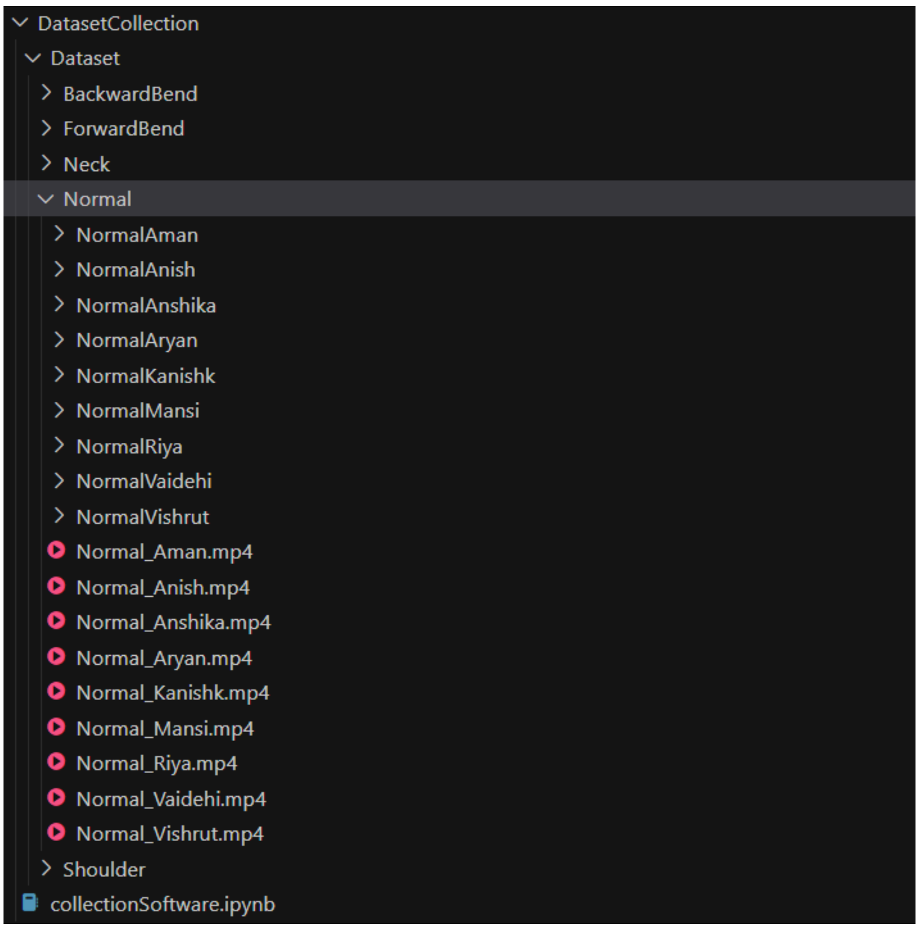

Figure 9. Structure Of The Raw Data Collected

Data Integration Analysis and Quality Evaluation

In this step, the structure of the collected dataset was changed to clearly divide our data into just 5 classes while combining the same posture frames of all the users.

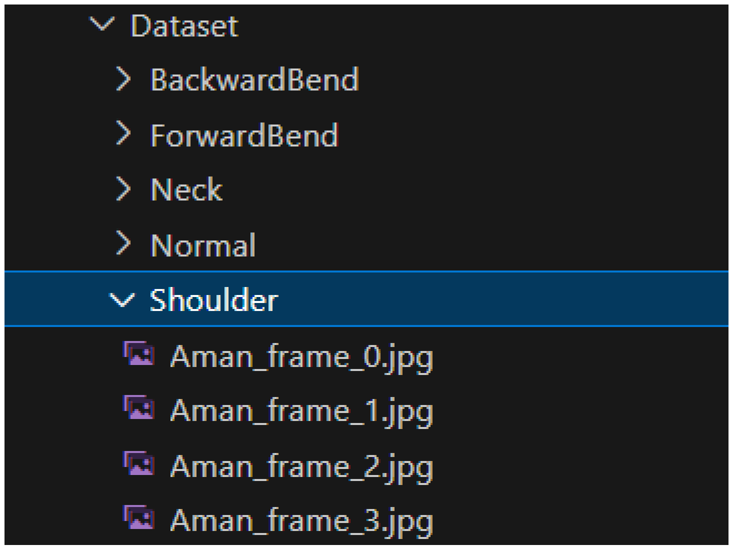

Figure 10. Final Structure Of The Dataset

Now, as we have the final dataset, it was meticulously analyzed and filtered so we have reliable data to train our model on. After this step, i.e, deleting the misclassified frames, we had our dataset ready for the data pre-processing step.

Data Pre-Processing

Data pre-processing involved several steps to make it easier for our model to train on these images. This steps included :

- Resizing all the frames to a constant size, i.e, 256X256 px.
- Then all the frames which were initially RGB(3 Channeled) were converted to Grayscale(1 Channel)
- Now, all the frames were normalized, so that pixel values lie only in the range 0-1 rather than 0-255.

In summary, our images which were initially of size (680, 480, 3) and with pixel values in range (0-255), were converted to a constant shape of (256, 256, 1) with pixel values in a minimized range (0-1)

```
def imageProcessor(image_path):
    image = cv2.imread(image_path)
    image = cv2.resize(image, (256, 256))
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = np.expand_dims(image, axis=-1)
    image = image.astype(np.float32) / 255.0
    return image
```

Figure 11. Data Pre-Processor

The above function was applied to all the 40K images in the dataset before feeding them into the model.

4.2.4 POSTURE DETECTION ALGORITHM

Introduction

This step involved the implementation of the backbone of our project that is the main algorithm which is responsible for detecting and classifying human postures as good/bad or posture specific classifications. Without the implementation of this step, the system makes no sense, so this was the most crucial, challenging yet important step of our project implementation pipeline. In the context of our project, this posture detection algorithm utilizes computer vision techniques and machine learning models to analyze images captured by a webcam or camera feed. The algorithm's aim is to accurately classify various postures based features extracted from the images.

Algorithm Overview

A Convolutional Neural Network was trained on the dataset to classify them into different postures. Several iterations and various model architectures were used to obtain a high level of accuracy and reliability. These iterations had a severe impact on choosing the right hyperparameters, data and classes for classification. We trained our CNN model using a large dataset of labeled posture images just collected from our own script. During the training process, the CNN learned to map input images to corresponding posture categories by adjusting the weights of its internal layers. CNNs flexibility and scalability allowed us to tailor the model architecture and dataset size to suit the complexity of our posture detection task. Additionally, overfitting and avoiding techniques offered by CNN like Dropout enabled us to enhance model's performance and robustness.

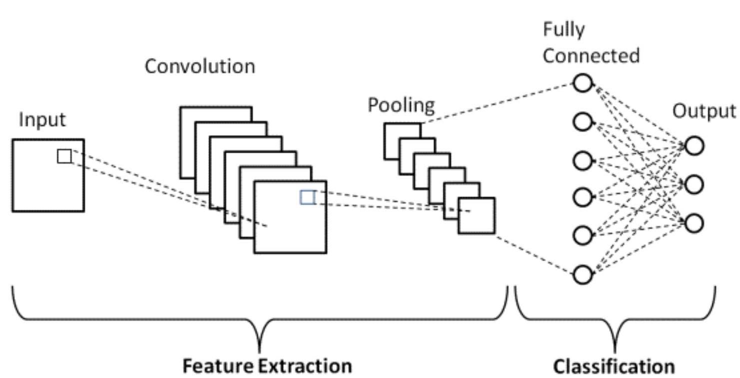

Figure 12. Basic CNN Architecture [2]

Model Architecture

- Input Layer (Conv2D):
    - Number of filters: 64 Filter size: (3, 3)
    - Activation function: ReLU
    - Input shape: (256, 256, 1) - This indicates grayscale images of size 256x256 pixels.
- Pooling Layer (MaxPool2D):
    - Pool size: (2, 2) This layer reduces the spatial dimensions of the input volume.
- Dropout Layer:
    - Rate: 0.25 Dropout is applied to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training.
- Convolutional Layer (Conv2D):
    - Number of filters: 128
    - Filter size: (3, 3)
    - Activation function: ReLU
- Pooling Layer (MaxPool2D):
    - Pool size: (2, 2)
- Dropout Layer:
    - Rate: 0.25
- Convolutional Layer (Conv2D):
    - Number of filters: 256
    - Filter size: (3, 3)
    - Activation function: ReLU
- Pooling Layer (MaxPool2D):
    - Pool size: (2, 2)
- Dropout Layer:
    - Rate: 0.25
- Flatten Layer: This layer flattens the input volume into a one-dimensional tensor to feed into the fully connected layers.
- Fully Connected Layer (Dense):
    - Number of neurons: 256

        ■ Activation function: ReLU This layer connects every neuron in the previous layer to every neuron in this layer.

- Dropout Layer:
    - Rate: 0.5
- Fully Connected Layer (Dense):
    - Number of neurons: 128
    - Activation function: ReLU
- Dropout Layer:
    - Rate: 0.5
- Output Layer (Dense):
    - Number of neurons: 3
    - Activation function: Softmax This layer produces the output probabilities for each class. Since there are 3 classes, the output layer has 3 neurons.

NOTE : It was observed that classes backward bend and misaligned shoulders were reducing the model's accuracy so they were removed from the dataset, leaving us with just 3 classes and 20k total dataset

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 254, 254, 64)      640
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 127, 127, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 127, 127, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 125, 125, 128)     73856
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 62, 62, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 62, 62, 128)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 60, 60, 256)       295168
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 30, 30, 256)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 30, 30, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 230400)            0
_________________________________________________________________
dense (Dense)                (None, 256)               58982656
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
=================================================================
Total params: 59,358,603
Trainable params: 59,358,603
Non-trainable params: 0
_________________________________________________________________
```

Figure 13. Model Architecture

Training Machine

Now we had to train our defined CNN model on our collected dataset. The CNN model we defined had a whopping of around 60 Million trainable parameters which simply tells how complex the model and the training process was. Huge size of the dataset, i.e, 20K images and each image that is 256x256, also played a very important role in making the training process a challenge.

Such a complex model was taking a huge time to train on our personal machines and that too when we reduced the batch size to as low as 1. On increasing the batch size, we received "memory limit exceed" errors. Even after using CUDA and CUDnn support for training tensorflow models on GPU, our personal machines failed to train even a single epoch of this model.

We had to train the model on the DGX Workstation to solve this issue. DGX workstation is a powerful computing facility provided by Nvidia that has 4 GPU's with several Cuda and Tensor cores per GPU along with a massive memory of around 128 GB. Only intricate task now was to upload this massive dataset on the DGX workstation which works via the internet. After uploading all the necessary files to the workstation, DGX was able to train and tune our model very efficiently.

DGX Workstation Specs :

- Processor : 20-core intel Xeon e5-2698 v4 2.2 GHz
- System Memory : 256 GB DDR4
- GPU : 4 x GPUs (Tesla V100 SXM2)
- GPU Memory : 128 GB (8X32) total system
- Cuda cores : 5000/GPU
- Tensor cores : 600/GPU
- Storage : 4X 1.92 TB SSD RAID 0
- OS : Ubuntu Linux

Training Process

In this process we fit our dataset onto our model and started with the training and tuning operations of this project. This was the most important step of the whole project. This involved several steps such as splitting the dataset into different sets, for example, training, testing and validation, assessing model outputs to determine accuracy, precision, etc and also employing cross validation and grid search for tuning the hyperparameters of the model.

Firstly the dataset was divided into training and testing sets while keeping the test size as 20%. Later on the training set was further divided using a 80:20 split to get our validation set.

After this step our sets looked like :

- Training set : 12800 images
- Testing set : 4000 images
- Validation set : 3200 images

Now the model was compiled and the training process was started while employing hyperparameter tuning techniques like cross validation and grid searching. A 5 fold cross validation was used in combination with grid search CV to find the optimal hyperparameters.

Final chosen hyperparameters for this model were :

- Batch Size : 128
- Number of Epochs : 10 (With early stopping to prevent overfitting)

Optimizer and loss function chosen to compile the model were :

- Optimizer = “adam”
- Loss = “Categorical_CrossEntropy”

Keras Callbacks used :

- Early Stopping

Now the training was started while keeping a check on how well the trained model is performing on the validation set to avoid overfitting. In conjunction with this technique, early stopping was also employed to test the model's accuracy on the testing set and stop training and get best model parameters if testing accuracy reduces for 3 epochs continuously.

```
Epoch 1/10
98/98 [==============================] - 8s 69ms/step - loss: 1.0908 - accuracy: 0.4423 - val_loss: 0.4466 - val_accuracy: 0.8772
Epoch 2/10
98/98 [==============================] - 6s 62ms/step - loss: 0.3960 - accuracy: 0.8223 - val_loss: 0.0645 - val_accuracy: 0.9837
Epoch 3/10
98/98 [==============================] - 6s 63ms/step - loss: 0.1573 - accuracy: 0.9220 - val_loss: 0.0309 - val_accuracy: 0.9904
Epoch 4/10
98/98 [==============================] - 6s 62ms/step - loss: 0.1111 - accuracy: 0.9508 - val_loss: 0.0169 - val_accuracy: 0.9958
Epoch 5/10
98/98 [==============================] - 6s 63ms/step - loss: 0.0831 - accuracy: 0.9624 - val_loss: 0.0074 - val_accuracy: 0.9984
Epoch 6/10
98/98 [==============================] - 6s 62ms/step - loss: 0.0710 - accuracy: 0.9651 - val_loss: 0.0059 - val_accuracy: 0.9987
Epoch 7/10
98/98 [==============================] - 6s 63ms/step - loss: 0.0691 - accuracy: 0.9657 - val_loss: 0.0030 - val_accuracy: 0.9994
Epoch 8/10
98/98 [==============================] - 6s 63ms/step - loss: 0.0655 - accuracy: 0.9687 - val_loss: 0.0016 - val_accuracy: 0.9997
Epoch 9/10
98/98 [==============================] - 6s 63ms/step - loss: 0.0569 - accuracy: 0.9691 - val_loss: 0.0016 - val_accuracy: 0.9997
Epoch 10/10
98/98 [==============================] - 6s 62ms/step - loss: 0.0585 - accuracy: 0.9702 - val_loss: 0.0033 - val_accuracy: 0.9994
```

Figure 14. Model Training

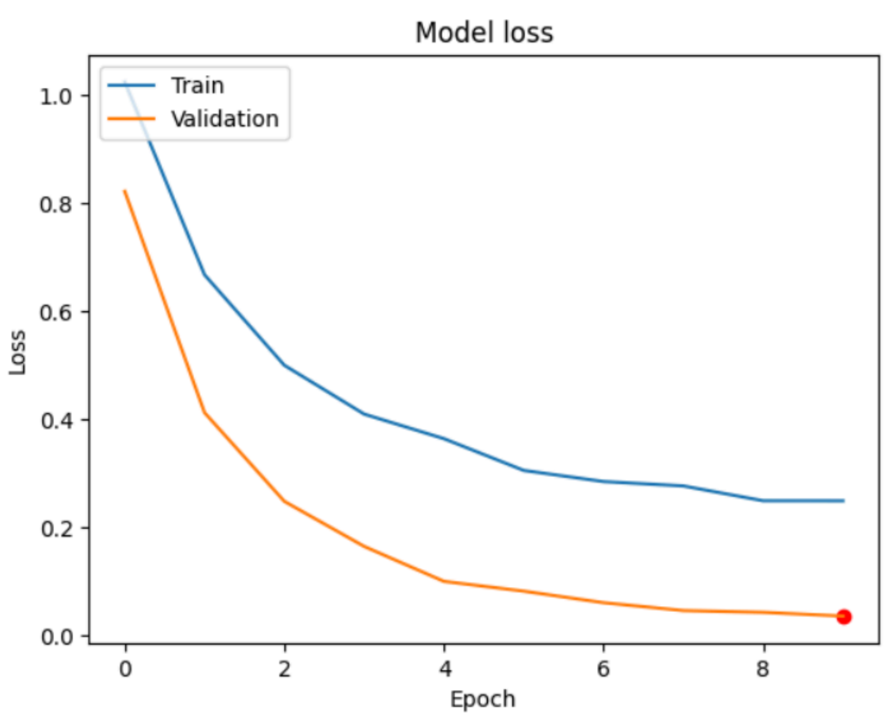

Figure 15. Model's Loss v/s Epochs Graph

Performance Evaluation, Inference & Classification

This step involves testing the model's inference on the unseen data, i.e, our testing dataset, new dataset collected through our collection software and the fresh data captured using embedded webcam.

This process was divided into three main parts :

1. First step was testing the model on the test dataset we split earlier. In this method the test dataset was loaded and pre-processed so as to fit our model required shape and output by the model were compared with the ground truth values which were the original labels of each image. This was done for our testing set of 3000 images and results were printed using a classification report. The classification report gave us insights about the model's precision, recall, F1 score and support.

Type	Precision	Recall	F1-Score	Support
Class Normal	0.96	1.00	0.98	1001
Class Neck	1.00	0.97	0.98	987
Class Forward Bend	1.00	0.99	1.00	1012
Micro Average	0.99	0.99	0.99	3000
Macro Average	0.99	0.99	0.99	3000
Weighted Average	0.99	0.99	0.99	3000
Samples Average	0.99	0.99	0.99	3000

Table 2. Testing Phase 1

2. In the second step of testing the model, a new dataset was collected from new volunteers who were not a part of the training data collection process. Through this method we were able to identify how well our model was performing on truly unseen data and that too collected in a newer environment.. This was done for a collected dataset of around 7000 images and results were printed using a classification report. The classification report gave us insights about the model's precision, recall, F1 score and support.

Type	**Precision**	**Recall**	**F1-Score**	**Support**
Class Normal	0.72	0.50	0.59	2400
Class Neck	0.61	0.57	0.59	1966
Class Forward Bend	0.60	0.78	0.68	2681
Micro Average	0.63	0.63	0.63	7047
Macro Average	0.64	0.62	0.62	7047
Weighted Average	0.64	0.63	0.62	7047
Samples Average	0.63	0.63	0.63	7047

Table 3. Testing Phase 2

3. In the third part of the inference and evaluation process, a python script was written to open the user's webcam and capture frames every 10 milliseconds. These captured frames were then processed to fit our model’s requirement and then the predictions were displayed on the user’s screen so that the user can manually assess the working and accuracy of the model.

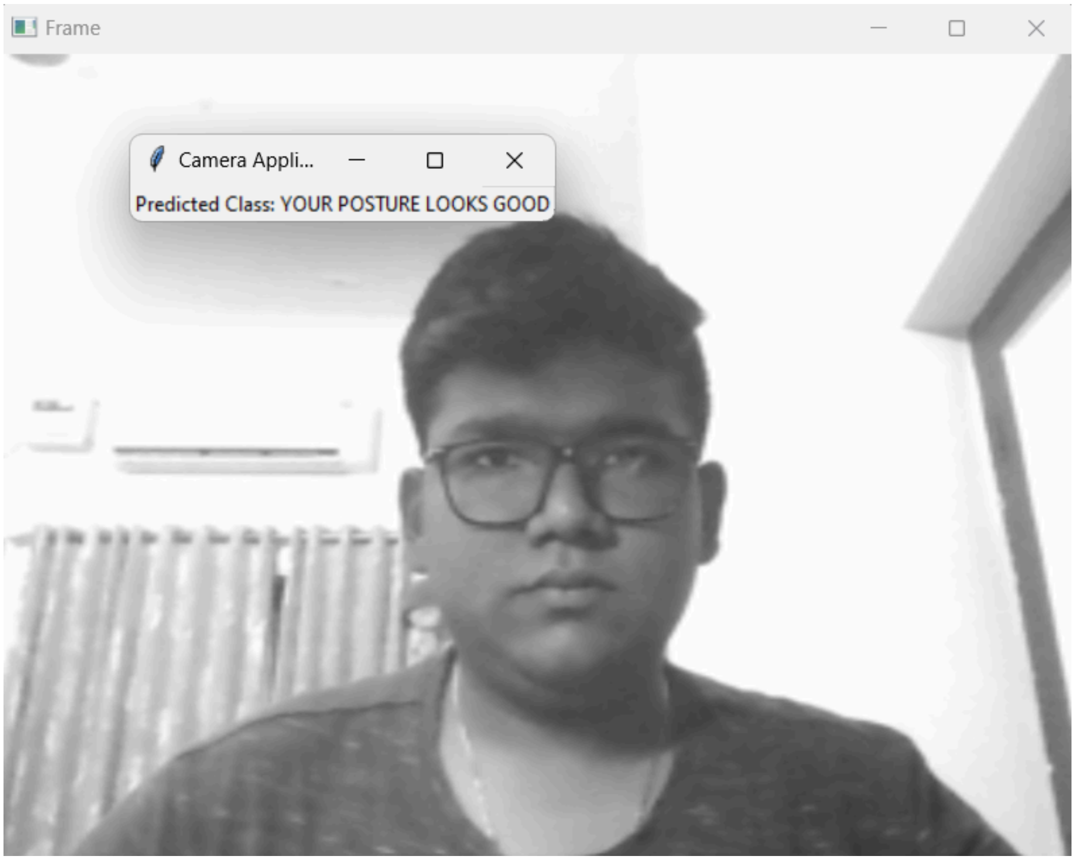

Figure 17. Inference App

This process also involved evaluating the speed and efficiency of the trained model. It was observed that the model took an average of 20ms of time to process and infer each frame.

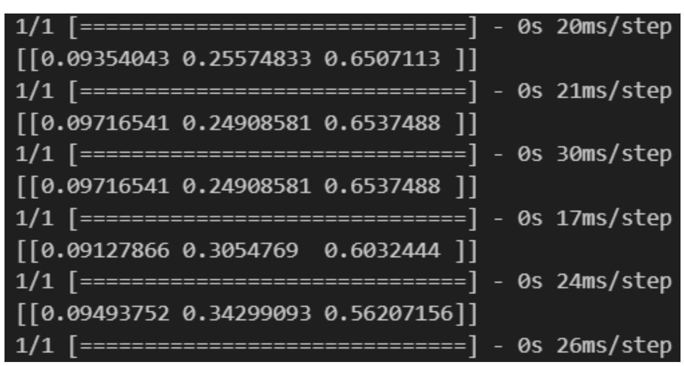

Figure 18. Inference Time

Post-Processing Steps

This step involved the post-processing which was required to retrieve and analyze the model outputs. Our model's output layer was a softmax layer and as a result, our model's output was an array of length three having probabilistic values for each class. Another factor which led to the addition of this step was the inference time of our model. To predict the classes in real time we were capturing frames at 10ms but frames were getting processed at 20ms, which led to delay in predictions which increased exponentially with time.

To conquer this issue some post processing steps were applied to the model's output and inference techniques were changed. Our new approach involved capturing frames every 20ms for a time period of 2 seconds and storing each prediction in an array. This step ensured that after every 2 seconds of time we had a total of 100 predictions in the array. To store the predictions, an additional condition was added that if the probabilistic value of predicted class is greater than 60% then append that class, otherwise, predict the class to be normal. This ensured that our model has higher chances of getting false negative rather than false positive which is better for our use case. Now that we have 100 predictions over a period of 2 seconds, the class which was predicted the highest number of times in these 2 seconds was considered to be the final prediction of the posture for the time frame of the last 2 seconds. This predicted class was then displayed to the user.

In conclusion, after adding this step to our pipeline, we modified the inference to work in real time but not every 10 ms but for every 2 seconds. We slowed the inference a little bit but still managed to boost our accuracy further with this because now rather than relying on 1 frame for every prediction, we used 100 frame predictions for our final inferred class.

4.2.5 USER INTERFACE

User Interface was the final yet a challenging phase of the development process. UI's only purpose was to display and showcase whatever was created before this step. The UI of the application would hold the majority factor on deciding whether the user will use our system or not. It was really important to keep the UI simple, delivering all our functionality while keeping it appealing and engaging for the user. UI was also meant to be kept easy to learn and user friendly for user's of diverse groups.

For building the user interface of our system, python's Tkinter library was used, which is a python wrapper for Tk GUI Toolkit. Its feasibility and support for vast operating systems made it a clear choice for our project. Before starting with the UI development for the system, user interface mockup diagrams were created for each page of the application. Once the modeling of the UI was completed, we took advantage of numerous features and widgets provided by Tkinter to successfully implement a professional UI for the system.

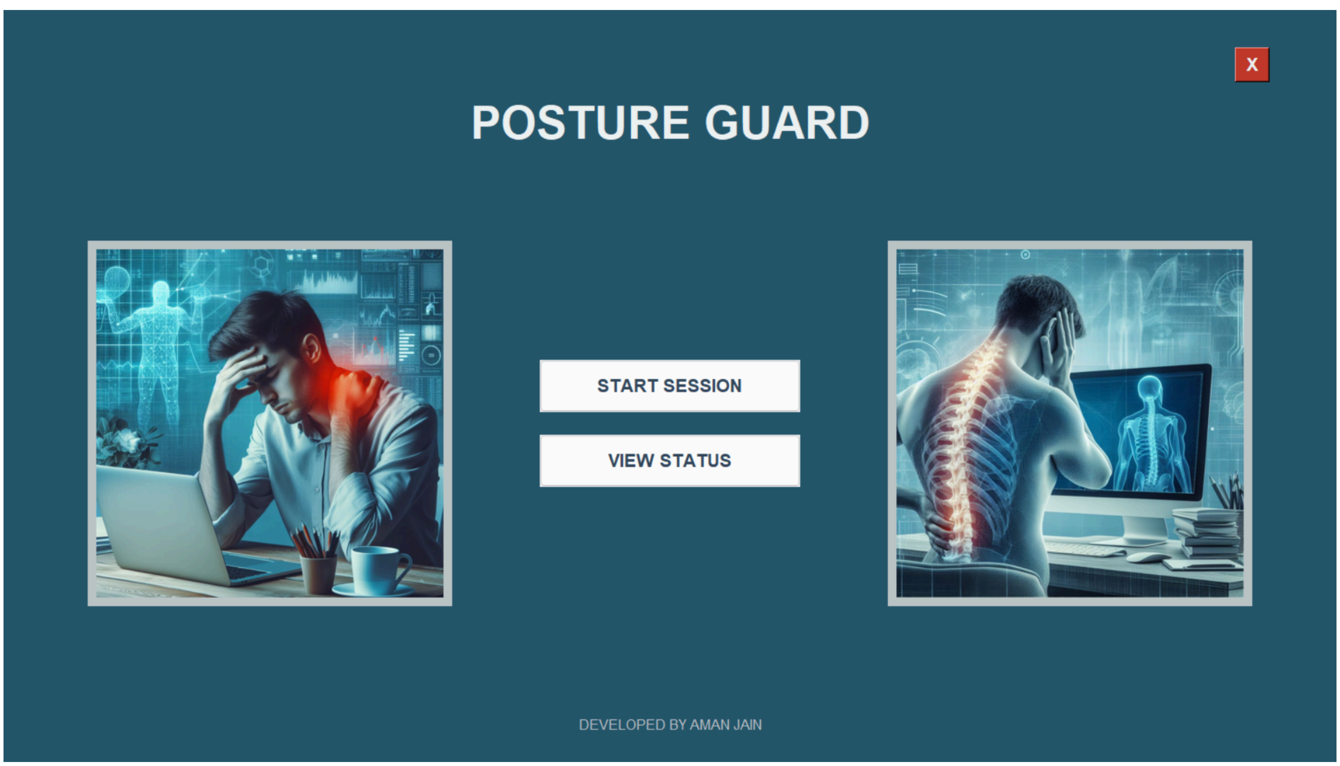

Figure 19. Home Page

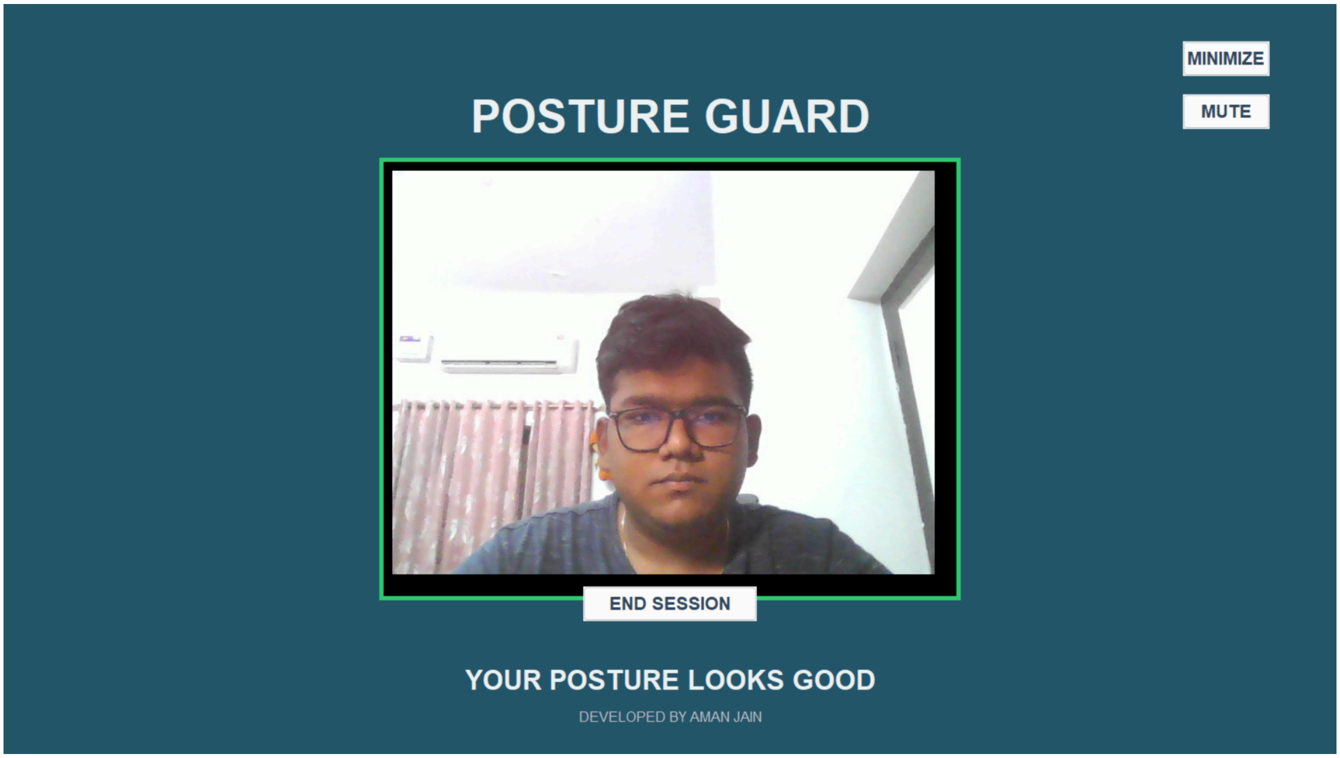

Figure 20. Session Page

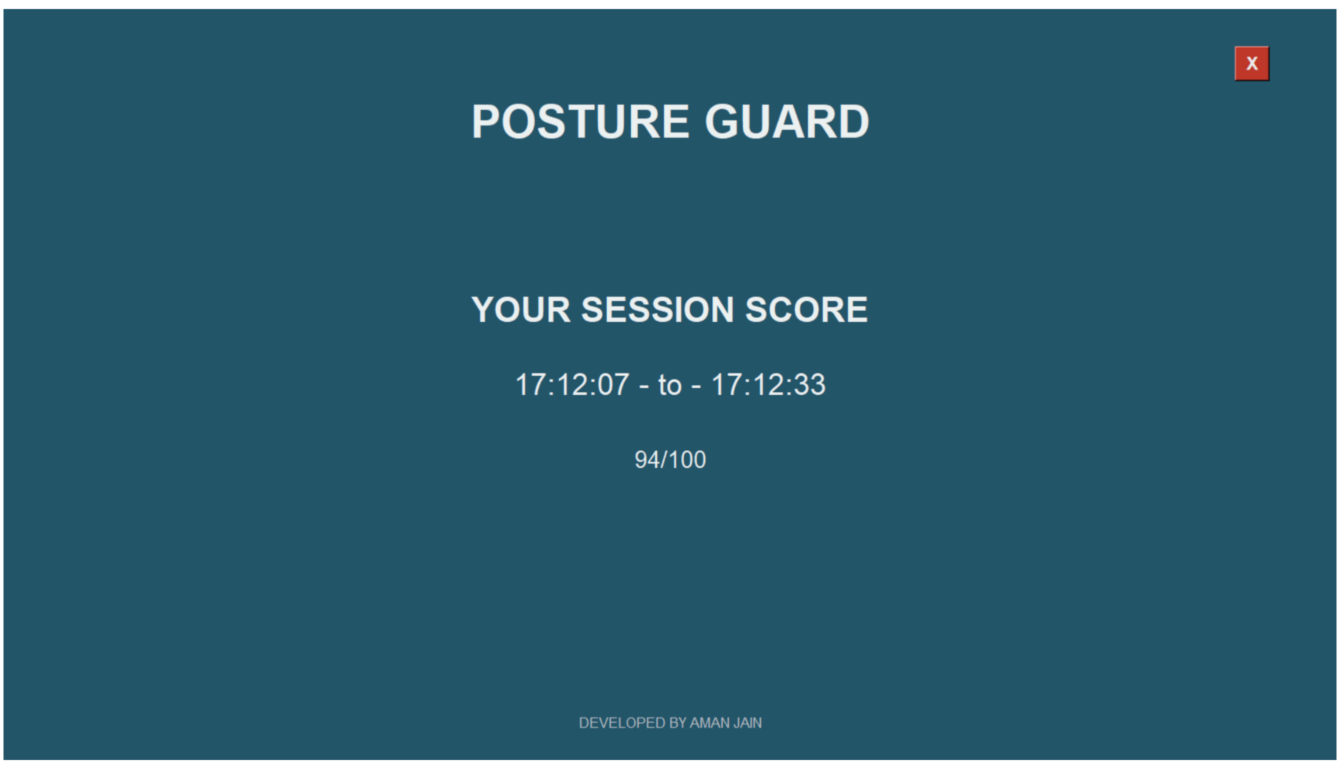

Figure 21. Status Page

Since, our system's main objective is to boost productivity and efficiency of a user by preventing health and musculoskeletal related pains and diseases, it is of no use that this app always runs in the foreground and restricts the users to perform their daily activities. It is of great importance that this app runs in the background and give pop up notifications to the user only when their posture is not optimal. This functionality was also implemented in Tkinter. A button named "MINIMIZE" was added on clicking which, the apps get minimized to the taskbar but it still runs, captures frames and predicts the posture. When the app is minimized, notifications come up as transparent pop ups which vanish automatically after 2 seconds. A sound alert feature was also implemented, which plays a beep sound on detecting suboptimal postures. User has the option to mute or unmute this sound alert feature as per his need.

Figure 22. Notification System

All these features make this system completely customizable by the users, in which they have a pool of options to choose from on how they want to get notified about their bad postures.

4.2.6 DEPLOYMENT

Introduction

Now, that our project is completed including the functions and algorithms to detect the suboptimal postures, user interface of the application and notifications, alerts and feedback mechanisms, there was now a need of deploying this system so that this can be brought to real world usage by users in different environments like workplaces, offices, schools, colleges and personal systems.

Deployment Strategy & Process

Deployment of this system was divided into two parts :

1. PyInstaller : PyInstaller played an instrumental role in the deployment of our system, providing us with a seamless way to package our Python application into a standalone executable for various platforms. We utilized PyInstaller to convert our Python codebase into an executable file that could be run on target systems without requiring Python interpreter installation. This simplified the deployment process for end-users, eliminating the need for manual installation of dependencies like Python and its libraries and ensuring compatibility across different operating systems. Additionally, PyInstaller offered options for customizing the executable file's appearance, including icon selection, enhancing the overall user experience. By making use of PyInstaller's capabilities, we were able to streamline the deployment process and deliver our system as a self-contained package to end-users, minimizing installation complexities and ensuring a smooth user experience.

2. INNO Setup : INNO Setup also played a crucial role in the deployment of our system by providing a robust and flexible toolset for creating Windows installers. We used INNO Setup to create installation packages(Windows Installer) that encapsulated our system's executable file, data files, and dependencies, along with necessary installation scripts and user interface elements. INNO Setup allowed us to customize the installer's appearance and behavior, including options for selecting installation components, defining installation paths and creating desktop shortcuts. By leveraging INNO Setup, we were able to create professional-looking installation packages for our system, simplifying the deployment process for end-users and

enhancing the overall user experience on the Windows platform. Another main reason for deploying our system using INNO setup was the compression it provided. Using PyInstaller alone gave us an executable file of size approximately 1.5 GB which was really difficult to distribute given its large size. On encapsulating this exe file using INNO into a windows installer, gave us an installer executable of size approximately 250 MB. It's clearly very less in size as compared to application executable which makes it very easy for distribution. This 250 MB installer is capable of installing our system as a Windows application on any system running on Windows. INNO also provided us with an easy uninstallation feature which would really help the user in case of bugs and errors.

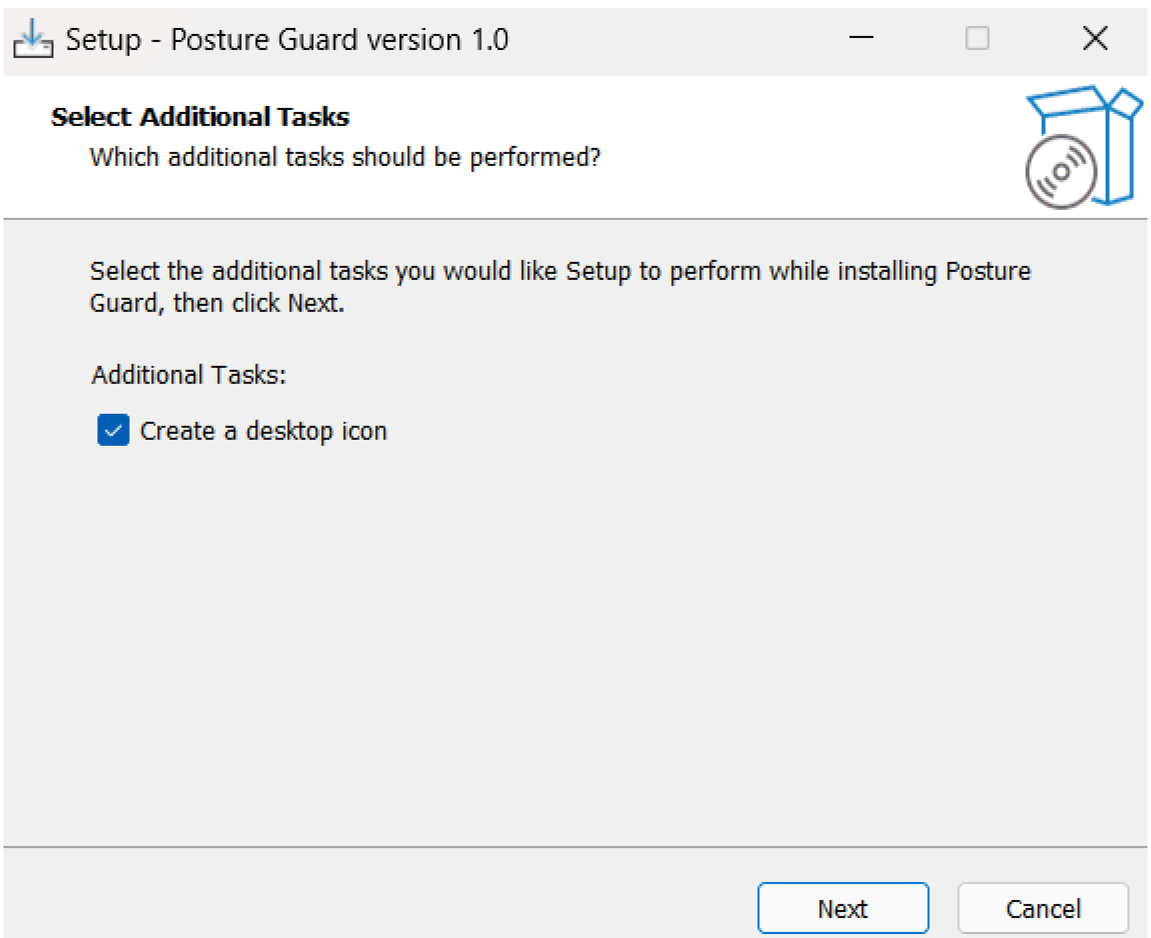

Figure 23. Posture Guard Installer

4.2.7 ISSUES AND CHALLENGES

Throughout the development phase of this project, several challenges came forward which were hard to tackle but we somehow managed to conquer them with thorough research and experimentation.
The challenges and issues we faced were :

1. **No availability of dataset :** Since, there was no dataset available for our use, it created a great delay in implementing this project because we had to create a data collection software and run it many times with diverse users for successful collection of good quality data. It was a time consuming process and resulted in delay since no other step could be implemented without the dataset.
2. **Training complexity of the CNN model :** Since, the CNN model architecture we defined was really complex with around 60 Million parameters and also the dataset on which it had to be trained on was huge with each image being 256x256 px and a total of 20K images, it became very hard to train such a giant model on our personal machines. It was taking nearly an hour to train only a single epoch and that too with a really low batch size. To conquer this challenge, we had to take help from the DGX workstation, which is a supercomputing facility provided by Nvidia.
3. **Real World Testing :** Once the system was developed completely, it was time to deploy this software in the real world for extensive testing in a diverse environment by a wide diversity of users. It was quite challenging as first we needed to educate people about postural health and then they would somehow agree to try and test our software once.
4. **Pushing Updates :** Whenever we find a need to change something whether in algorithm or the user interface, it is quite challenging to push them on the end user's system since our app is not hosted anywhere online from where updates can be pushed directly. Only method to install updates is to install the application with updates installer file.

4.3 RISK ANALYSIS AND MITIGATION

There were several risks identified during the course of this project. Extensive research and experimentation helped us to mitigate those risks for betterment of this project.

- Technical Risks: Technical complexities in implementing computer vision algorithms and integrating them into the user interface.
    - Mitigation: Conducted thorough research and prototyping to understand the feasibility and challenges of the chosen algorithms. Allocated sufficient time for testing and debugging.
- Hardware and Software Dependencies Risk: Dependency on specific hardware components like embedded camera or software libraries that may have compatibility issues or limited support.
    - Mitigation: Prioritized the use of widely supported hardware and software platforms. Maintained clear documentation of all dependencies and their versions.
- Data Privacy and Security Risk: Risks associated with handling sensitive user data, such as personal images or health information, leading to privacy breaches or data leaks.
    - Mitigation: Implemented system in such a way that id doesn't store any data for a longer period of time. User's images are deleted after 2 seconds.
- User Acceptance and Usability Risk: Users may find the system interface complex or unintuitive, leading to low adoption rates or user dissatisfaction.
    - Mitigation: Prioritized user-centered design principles and conducted usability testing with target users to gather feedback and refine the user interface.

CHAPTER 5

TESTING

5.1 TESTING PLAN

Testing plan for our project included several different types of testing :

- **Unit Testing:** Tested individual components of our system such as the posture detection algorithm, image preprocessing, feature extraction, and classification, to ensure they function correctly.
- **Integration Testing:** We also tested the integration of different modules involved in the system, including the posture detection algorithm, user interface, and audiovisual feedback mechanism. Verified that data flows smoothly between modules and that inputs and outputs are processed correctly.
- **Functional Testing:** Tested the overall functionality of our system by simulating user interactions with the user interface. Verified that the system is accurately detecting and classifying different postures in real-time. Tested various scenarios, such as different lighting conditions, backgrounds, and user positions, to ensure robustness and accuracy.
- **Performance Testing:** Measured the performance of the system in terms of speed, accuracy, and resource utilization. Evaluated the processing time required for posture detection and classification.
- **Usability Testing:** Conducted usability testing with representative users to evaluate the user interface design and overall user experience. Gathered feedback on the clarity of instructions, ease of use, and intuitiveness of the interface. Identified any usability and made necessary adjustments to improve user satisfaction.

By following this testing plan, we systematically validated and verified the functionality, performance and usability of our posture detection system, ensuring its reliability and effectiveness in the real world scenarios.

5.2 LIST OF TEST CASES

TEST CASE 1	VERIFIED THAT IMAGES ARE BEING PRE-PROCESSED CORRECTLY
TEST CASE 2	VERIFIED THAT KNOWN IMAGES ARE BEING CLASSIFIED CORRECTLY
TEST CASE 3	TESTED THE MODEL'S PERFORMANCE ON UNSEEN IMAGES
TEST CASE 4	TESTED THE MODEL'S CAPABILITY TO HANDLE DIFFERENT POSTURE VARIATIONS
TEST CASE 5	TESTED WHETHER THE SYSTEM CAN CLASSIFY LIVE VIDEO STREAMS IN REAL TIME
TEST CASE 6	TESTED THE MODELS RESPONSE TO REAL TIME CHANGES IN POSTURES
TEST CASE 7	VERIFIED THE CORRECTNESS OF LIVE VIDEO STREAM DISPLAYED ON THE UI
TEST CASE 8	TESTED USER INTERACTIONS WITH THE INTERFACE ELEMENTS (BUTTONS)
TEST CASE 9	VERIFIED THAT THE PREDICTED CLASS IS DISPLAYED CORRECTLY ON THE UI
TEST CASE 10	TESTED THAT THE AUDIO FEEDBACK IS TRIGGERED TIMELY ON DETECTION OF BAD POSTURE
TEST CASE 11	VERIFIED THE CLARITY AND VISIBILITY OF VISUAL FEEDBACK

Image test cases :

A : Normal Posture

B : Neck Strain

C : Spine Strain

D : Face not in frame

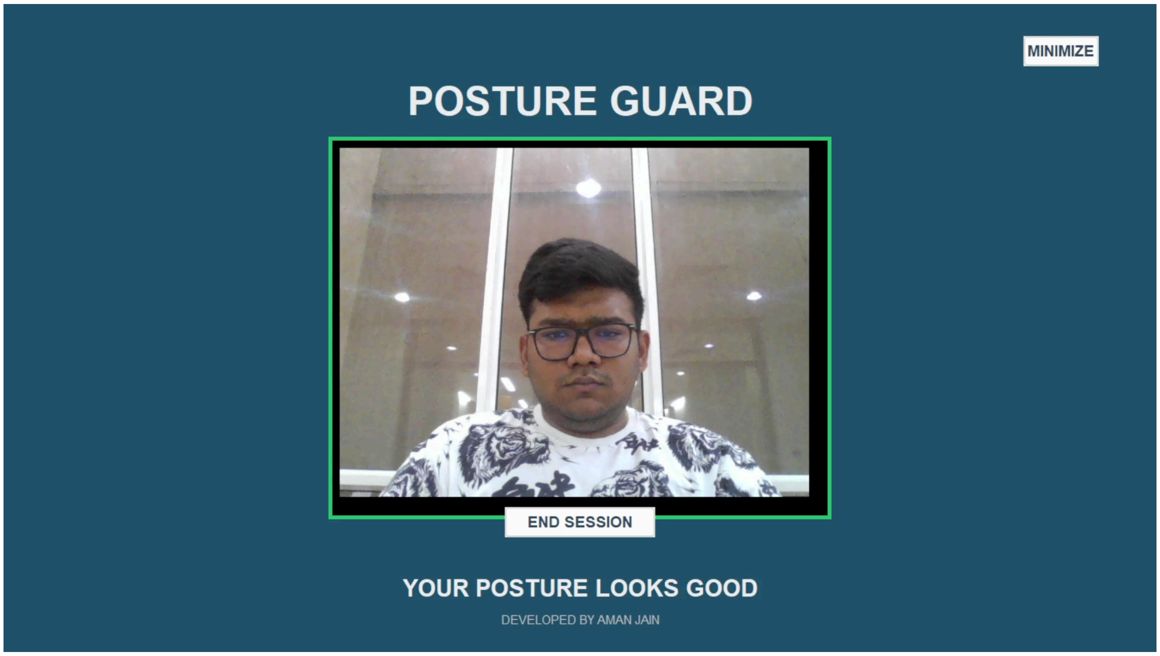

Figure 24. Image Test Case A

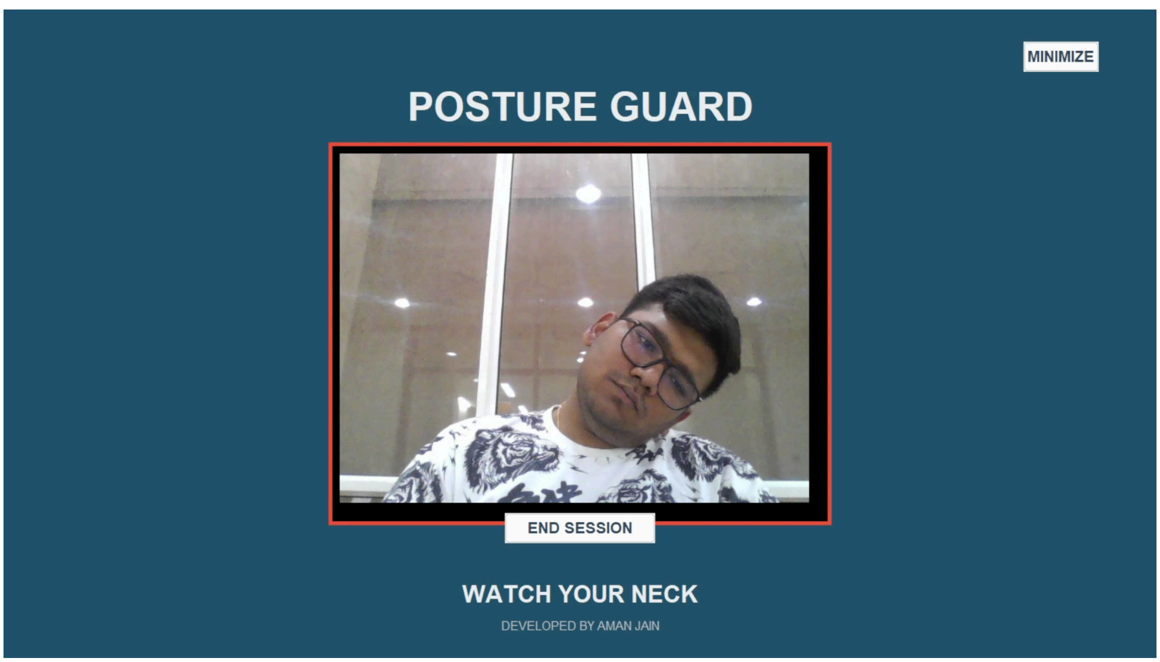

Figure 25. Image Test Case B

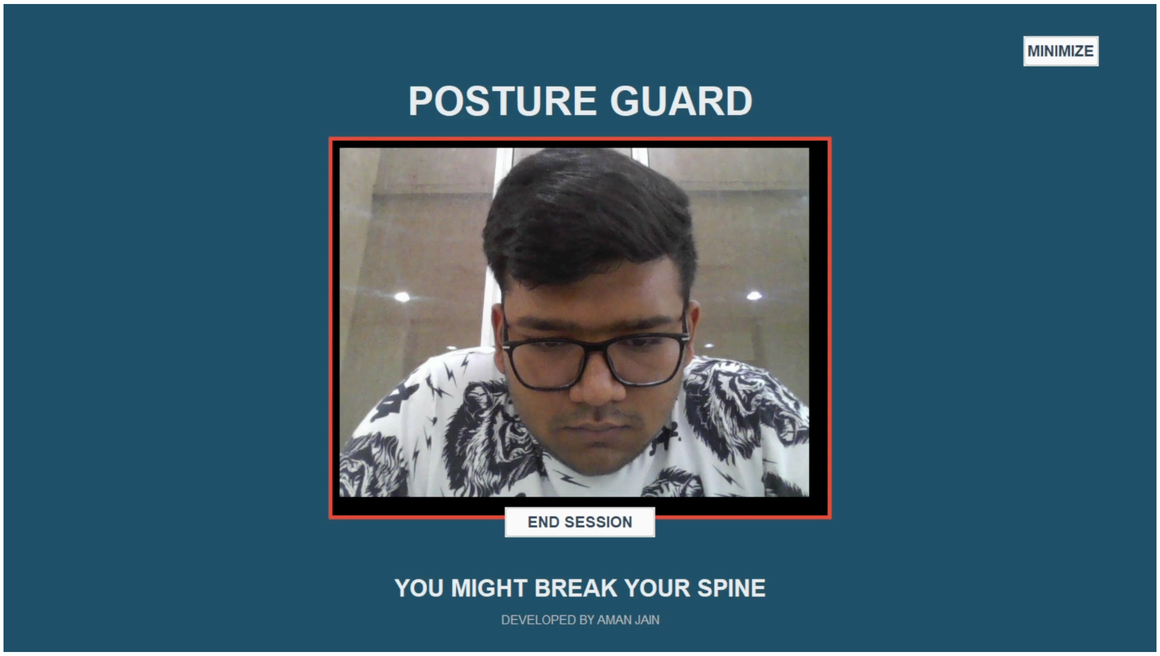

Figure 26. Image Test Case C

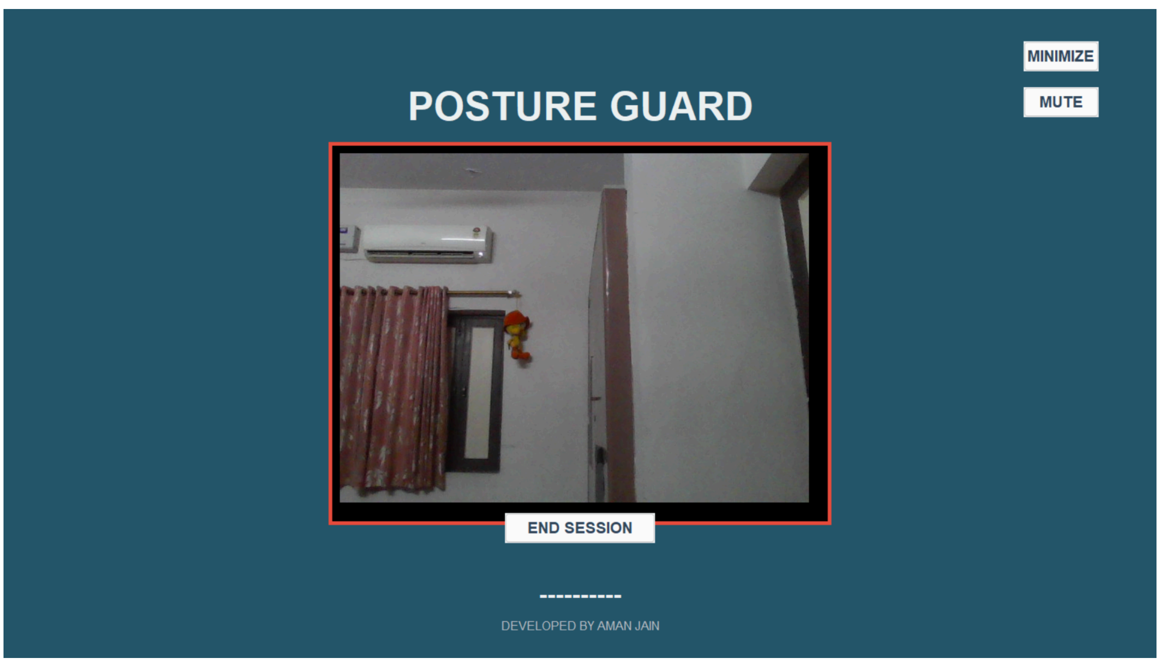

Figure 27. Image Test Case D

5.3 LIMITATIONS OF THE SOLUTION

While the system fulfills what it aims for in most of the scenarios, there are quite a few limitations to the solution approach we used here.

Here are some limitation which were identified during and post implementation of this project :

1. **Limited Posture Classes :** Currently we are training the model for only three specific classes, i.e, “Normal”, “Neck Strain” and “Back Strain”. This is a major limitation of the solution approach since users can opt for an infinitely large number of postures while using computer devices whereas our prediction algorithm only takes into account three postures. Yes, these three positions taken by us are major concerns but still there is a scope of improvement.
2. **Single Viewpoint :** Since our system is not using any special hardwares like smart sensor embedded chairs or depth sensing cameras, the whole working of our system relies on the computer’s inbuilt or attached webcam. Therefore, the positioning of the webcam plays a very vital role in determining that the system would work correctly or not. Since, every user has his/her own style of computer so their camera positions and angles won’t be the same, making the system unreliable in many cases.
3. **Limited Training Data :** Through our data collection software we have collected around 40K images but still this data is very less to train a robust model for such an intrinsic task of posture classification. Though the data is as large as 40,000 images, it was collected by only 9 people. For such a problem statement which we are working on, the dataset needs to be as diverse as possible. Adding people from different gender, race, color and clothes would help us improve the model accuracy tremendously,
4. **Environmental Variability :** This limitation is another outcome of a limited dataset. Since our dataset was collected only 9 times, we weren’t able to accommodate very diverse environment scenarios like lighting, darkness, crowd around the person using the computer, etc.

Acknowledging these limitations allows us for a more comprehensive understanding of the posture detection solution's capabilities and potential areas for improvement and refinement.

CHAPTER 6

CONCLUSION & FUTURE WORK

6.1 CONCLUSION

In conclusion, the posture detection system presented in this project demonstrates significant potential in addressing the growing concerns surrounding poor posture habits and their impact on musculoskeletal health. Through the integration of computer vision techniques, machine learning algorithms, and user-friendly interfaces, the system is offering an effective solution for real-time posture monitoring and feedback. By leveraging Convolutional Neural Networks (CNNs) and advanced image processing algorithms, the system achieves robust posture classification accuracy, enabling users to receive timely alerts. Furthermore, the deployment of the system through user-friendly interfaces, such as a desktop application, enhances accessibility and usability across diverse user demographics.

The successful implementation and evaluation of the system underscore its practical viability and effectiveness in promoting postural awareness and healthy habits among users. Moreover, the successful implementation of this project shows its applicability across various domains, including healthcare, ergonomics, and workplace safety. The user-centric design and intuitive interfaces ensure seamless integration into users' daily routines, fostering greater awareness and adoption of healthy postural habits. As a result, the posture detection system holds promise not only in mitigating the adverse effects of poor posture but also in promoting overall musculoskeletal wellness and enhancing quality of life for individuals across diverse demographics and settings.

6.2 FUTURE WORK

While the current posture detection system represents a significant advancement in the field, several opportunities for future research and development exist to enhance its capabilities and address emerging challenges.

Some potential areas for future work include:

- Personalized Recommendations: Incorporating machine learning models to personalize posture correction recommendations based on individual user characteristics, preferences, and historical data, thereby enhancing user engagement and adherence.
- Long-Term Monitoring: Extending the system's capabilities for long-term posture monitoring and trend analysis, enabling users to track their postural habits over time and identify potential risk factors for musculoskeletal disorders.
- Accessibility and Inclusivity: Ensuring the accessibility and inclusivity of the system for users with diverse abilities and needs, including support for adaptive interfaces, assistive technologies, and multi-language support.
- Clinical Validation: Conducting clinical studies and validation trials to assess the system's effectiveness in clinical settings and its impact on mitigating musculoskeletal disorders.

By pursuing these avenues for future work, the posture detection system can evolve into a comprehensive solution that not only detects and corrects poor posture habits but also promotes long-term musculoskeletal health and well-being among users.

APPENDIX A
"USER SURVEY"

Survey Title: Posture Guard Survey
Survey Date: 20th February 2024

Survey Questions:

- **Question 1:** Name
- **Question 2:** Gender
- **Question 3:** Age
- **Question 4:** How much time do you spend working on a computer daily?
  - 1-3 Hrs
  - 3-6 Hrs
  - 6-9 Hrs
  - 10+ Hrs
- **Question 5:** How often do you keep a check on your posture while working on a computer?
  - Never
  - Rarely
  - Occasionally
  - Frequently
  - Always
- **Question 6:** How crucial do you consider maintaining proper posture to be, on a scale of 1 to 10?
-

Note: This appendix provides an overview of the survey questions and options used in gathering user feedback for the project. Actual responses and analysis are detailed in the main body of the report.

REFERENCES

[1]Byeon, Y. et al. (2020). "Posture Recognition Using Ensemble Deep Models under Various Home Environments." *Applied Sciences.*

[2]Roseline, O. et al. (2020). "Human Posture Detection Using Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms." *MDPI Journals.*

[3]Audrius, K. et al. (2021). "Detection Of Sitting Posture Using Hierarchical Image Composition And Deep Learning." *PeerJ Computer Science Journals.*

[4]Enrique, P. et al. (2021). "A Deep-Learning Based Posture Detection System for Preventing Telework-Related Musculoskeletal Disorders." *MDPI Journals.*

[5]Devaraj, M. et al. (2023). "Modelling Proper and Improper Sitting Posture of Computer Users Using Machine Vision for a Human–Computer Intelligent Interactive System during COVID-19." *Applied Sciences.*

[6]Estrada, J. et al. (2023). "Computer Users Sitting Posture Classification Using Distinct Feature Points and Small Scale Convolutional Neural Network for Humana Computer Intelligent Interactive System During COVID-19." *Journal of Computer Science*

[7]Schmid, M. et al. (2023) "Recurrent Network Solutions for Human Posture Recognition Based on Kinect Skeletal Data." *Sensors.*

[8]Isaac, M. et al. (2024) "A Method For Complex Posture Recognition During Long-Term Sitting Using Neural Networks And Pressure Mapping Systems." *Science Direct*

20103236 Posture Guard : A Posture Detection & Alarming System

ORIGINALITY REPORT

13%	11%	9%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	4%
2	dr.ntu.edu.sg Internet Source	1%
3	Isaac Morales-Nolasco, Sandra Arias-Guzman, Laura Garay-Jiménez. "A method for complex posture recognition during long-term sitting using neural networks and pressure mapping systems", Biomedical Signal Processing and Control, 2024 Publication	1%
4	www.ncbi.nlm.nih.gov Internet Source	1%
5	shodhganga.inflibnet.ac.in Internet Source	1%
6	www.coursehero.com Internet Source	1%
7	open-innovation-projects.org Internet Source	1%

8	"Front Matter", 2023 8th International Conference on Computer Science and Engineering (UBMK), 2023 Publication	<1%
9	www.researchgate.net Internet Source	<1%
10	www.ijnrd.org Internet Source	<1%
11	de.slideshare.net Internet Source	<1%
12	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
13	ir.juit.ac.in:8080 Internet Source	<1%
14	medium.com Internet Source	<1%
15	etds.lib.ncku.edu.tw Internet Source	<1%
16	Submitted to M S Ramaiah University of Applied Sciences Student Paper	<1%
17	Roseline Oluwaseun Ogundokun, Rytis Maskeliūnas, Robertas Damaševičius. "Human Posture Detection Using Image Augmentation and Hyperparameter-	<1%

	Optimized Transfer Learning Algorithms", Applied Sciences, 2022 Publication	
18	discovery.researcher.life Internet Source	<1%
19	Submitted to Liverpool John Moores University Student Paper	<1%
20	www.jiit.ac.in Internet Source	<1%
21	Jianye Yuan, Song Li. "OMOFuse: An Optimized Dual-Attention Mechanism Model for Infrared and Visible Image Fusion", Mathematics, 2023 Publication	<1%
22	arxiv.org Internet Source	<1%
23	lrcdrs.bennett.edu.in Internet Source	<1%
24	"Ambient Intelligence – Software and Applications – 14th International Symposium on Ambient Intelligence", Springer Science and Business Media LLC, 2023 Publication	<1%
25	Submitted to University of West London Student Paper	<1%

26	Submitted to University of Strathclyde Student Paper	<1%
27	Submitted to Queen Mary and Westfield College Student Paper	<1%
28	Vaibhav Gulati, Srijan Dwivedi, Deepika Kumar, Jatin Wadhwa, Devaansh Dhingra, Jude D. Hemanth. "An automated framework to evaluate soft skills using posture and disfluency detection", Machine Vision and Applications, 2023 Publication	<1%
29	ia601300.us.archive.org Internet Source	<1%
30	www.lnmiit.ac.in Internet Source	<1%
31	Submitted to Birkbeck College Student Paper	<1%
32	Submitted to Icon College of Technology and Management Student Paper	<1%
33	Jheanel E. Estrada, Larry A. Vea, Madhavi Devaraj. "Modelling Proper and Improper Sitting Posture of Computer Users Using Machine Vision for a Human–Computer	<1%

Intelligent Interactive System during COVID-19", Applied Sciences, 2023
Publication

34	Submitted to University of Westminster Student Paper	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		

AMAN JAIN

Dehradun •India• amanjn2003@gmail.com • +91 9068870406 • [GitHub](#) • [Linkedin](#)

OBJECTIVE

Passionate Computer Science enthusiast with a keen interest in cutting-edge technologies, including Artificial Intelligence and Machine Learning. Eager to apply my skills and knowledge across various domains within computer science, including web development, app development, and beyond. Dedicated to leveraging my adaptable mindset and problem-solving abilities to tackle diverse challenges and contribute meaningfully to projects spanning different areas of technology. Committed to continuous learning and growth to excel in any aspect of computer science, embracing new challenges as opportunities for innovation and advancement.

SKILLS & INTERESTS

Technical Skills: Python, Machine Learning, Deep Learning, Computer Vision, Natural Language Processing, Git, Google Cloud, Web Development, SQL, C++, HTML, CSS, JavaScript

Soft Skills : Public Speaking, Leadership, Management, Marketing

Interests: Photography, Adobe Creative Suite

EDUCATION

Jaypee Institute of Information Technology — Noida, India

B.Tech, Computer Science — September 2020 - June 2024

GPA: 7.4/10

Relevant Coursework

Software Development Fundamentals • Data Structures • Algorithms • Statistics • Software Engineering • Computer Networks • Soft Computing • Natural Language Processing • Deep Learning • Big Data • Data Analysis

EXPERIENCE

Pathshala — Remote

Founder — March 2024 - Present

- Pathshala is a remote educational firm which provides dynamically curated courses to students.
- Continuously providing lectures and mentorships to many students.
- Looking after the marketing and management of the remote firm.

AI/ML Hub of JIIT — Noida, India

Founder/Senior Advisor — January 2024 - Present

- Founding member of the first hub related to Artificial Intelligence & Machine Learning at Jaypee Institute of Information Technology.
- Have participated in giving one on one training sessions to juniors about AI and ML.
- Organized various events serving more than 100 participants.

Tally Solutions Pvt. Ltd. — Bengaluru, India

Machine Learning Intern — May 2023 - July 2023

- Participated in a research project that entailed conducting an extensive literature review and meticulous documentation.
- Reviewed numerous research papers to develop and evaluate four innovative approaches for problem-solving.
- Extensively utilized the Google Cloud Platform to seamlessly integrate AI capabilities as an offline service.

PERSONAL PROJECTS

Posture Guard : Bad Posture Detection and Alarming System Present
Technologies used : Python • Computer Vision • Deep Learning

- Created a dataset collection software that can capture and annotate around 3000 images into five different classes in around 3 minutes of time.
- Carried out data pre processing for the MPII Human Pose Dataset.
- Training and testing various Convolutional Neural Network (CNN) architectures like HourGlass and UNet for the human pose estimation task.
- Deployed a windows application named Posture Guard for detecting and alraming the user of their bad postures while using computer devices.

Protein Structure Prediction December 2023
Technologies used: Python• Deep Learning• LSTM • XGBoost

- Applied intensive data pre processing on the protein structure dataset collected from protein data bank(PDB).
- Trained and tested various machine learning models like Long-Short Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost) to predict the protein structure from amino acid sequences.

Hybrid Approach to Solve Sudoku Puzzle January 2023
Technologies used : Python • Computer Vision • Deep Learning

- Integrated computer vision and deep learning techniques to engineer an optimized algorithm for automating the process of solving Sudoku puzzles.

Optical Character Recognition - Devanagari Lipi November 2022
Technologies used : Python • Deep Learning • CNN

- Designed and trained a Convolutional Neural Network (CNN) to recognize handwritten Hindi alphabets and numerals.
- Achieved an impressive accuracy rate of 98%.

Movie Recommendation System October 2022
Technologies used : Python • Sklearn

- Leveraged content-based filtering and a cosine similarity model to develop an efficient movie recommender system on a dataset containing 30,000 movies.

2048 Game January 2022
Technologies used : Web Development • HTML • CSS • JavaScript

- Utilized web development concepts, including HTML, CSS, and JavaScript, to create a functional web application for the popular game 2048.

Stone Paper Scissors Game November 2021
Technologies used : Web Development • HTML • CSS • JavaScript

- Utilized web development concepts, including HTML, CSS, and JavaScript, to create a play with computer stone paper scissors game .

ACHIEVEMENTS

Amazon ML Hackathon 2023 April 2023

- My team "Bit Theory" secured 93rd position amongst the 5000 participating teams.
- Collaborated with a team of four, to train a LSTM model on the Amazon products dataset.
- Had the privilege to improve my collaboration and leadership skills in this 3 day long hackathon.