Artificial Intelligence Posture Correction System

Daniel Chin
Computer Science and Engineering
University of California, Irvine
Irvine, United States
dpchinn@uci.edu

Syed F Ali
Computer Science and Engineering
University of California, Irvine
Irvine, United States
sfali1@uci.edu

Gaotong Yin
Electrical Engineering and Computer Science
University of California, Irvine
Irvine, United States
gaotongy@uci.edu

Ethan Hickey
Computer Engineering
University of California, Irvine
Irvine, United States
hickeye@uci.edu

Yanning Shen
Electrical Engineering and Computer Science
University of California, Irvine
Irvine, United States
yannings@uci.edu

I. Introduction

This project revolves around the development of **Posture Correction System powered by Artificial Intelligence (AI)**. The AI posture correction system is designed to address a variety of issues related to poor posture. It focuses on key areas such as reducing physical pain, preventing long-term health issues, improving physical appearance, and enhancing work efficiency [1]. By helping individuals maintain proper alignment, the system plays a crucial role in reducing physical discomfort and pain that often arises from chronic poor posture, particularly in areas like the neck, back, and shoulders. Furthermore, there is a growing trend towards integrating posture correction AI into daily workout routines [2]. By providing real-time feedback and guidance, the posture correction system can help users adjust their movements to avoid injury and maximize the benefits of their fitness regimen.

Our AI Posture Correction System wants to solve the following problems:

[RQ1]How do we extract skeleton data from an image?

[RQ2]How do we determine whether the posture in the image is correct or needs improvement?

[RQ3]How do we provide real-time feedback for detected abnormalities or discrepancies?

[RQ4]How do we design a user interface that is useful for end users as well as informational when evaluating performance?

II. Objectives

A. Overall Objectives

The potential benefits of such a tool are immense, ranging from **enhancing physical health to preventing chronic musculoskeletal issues** (akin to Arthritis and often caused by poor posture). The program is very helpful for people who have chronic scoliosis or lumbar spine injuries due to incorrect sitting posture habits [1]. This will enable them to learn about and improve their bodies based on the **feedback** provided by the AI engine.

B. Objectives for Fall Quarter 2023

Our Fall quarter was allocated for building the **initial framework** of our software (both front-end and back-end). Heavy research went into building our first machine learning model as well as structuring our datasets to provide meaningful categorical data. In addition to this, building the user interface for our software was integral to building a usable front-end to test our program on. The goal for the Fall quarter was to have **a working prototype** capable of recognizing when a video feed is showcasing good/bad posture and providing context as to why it is incorrect.

C. Objectives for Winter Quarter 2024

For the Winter quarter, our goal shifted from producing a working prototype to **perfecting** it. We began **testing different concepts** in AI algorithms in order to determine which method will most accurately estimate posture. Extensive research has gone into **applying different models** to our data and reorganizing our program to work as designed. Moreover, we transitioned from a web-based program to an embedded system to widen the scope of our project's usable scenarios. This required the Posture Correction System to be built using a Raspberry Pi, an LED screen, and a camera module to allow our program to run completely independently from a computer, allowing it to be placed anywhere for around-the-clock live posture analysis.

TABLE I
Completed Objects

Prototype	Months
Calendar Exercise Planner	September-December
Front end system design	November-January
Machine Learning model detect good and bad posture	November-February
Real-time Feedback	February-March
Get a working prototype on Raspberry Pi 5	February-March
Link Raspberry Pi to touchscreen and web camera	March

III. Setup Details

A. Gathering Datasets

The team comprises four computer scientists/software engineers, implying posture analysis is not the field of study. Thus, gathering a meaningful understanding of posture recognition and creating datasets of good and bad examples are the first priority [3]. Moreover, the datasets will be crucial in the training stage of the AI model, as examples must be given to help the machine learning model discern between good and bad posture.

To begin, a dataset of about 90 postures was trained over several models to strengthen the understanding of the discrepancies that need to be aware of [4]. With this start, the team can then create a dataset of students, instructing the program to take pictures in different postures and widening the scope of available data. Towards the Winter quarter, our team managed to collect over 200 unique postures to strengthen the dataset and fine-tune the models to predict more accurately.

B. Software Used

The project's software foundation is built on 5 different platforms. The AI model is written in **Python (v3.9)** and implements three major modules: Mediapipe, OpenCV, and Scikit-learn. The front-end software is entirely coded in **Python (v3.11)** as well.

- **Mediapipe (v0.10.8)** is responsible for reading video feeds from web cameras and transmitting that data into the program.
- **OpenCV (v4.9.0)** is the engine behind the skeleton estimation algorithms. It allows estimating the bone structure of a given person and returning data points in the form of 3D coordinates to process as data.
- **Scikit-learn (v1.2.2)** is the platform to build the machine learning platform on and will provide functions and algorithms to optimize data for machine learning.
- **PYQT 5 (v6.6.1)** is used to build the GUI aspect of the program and build a bridge between the back-end and front-end code.

C. Hardware Used

- **Desktop PC**: Used to develop the machine learning model and GUI. Also used for testing specific components during development.
- **Raspberry Pi 5 (6.1 Kernel)**: Operates on a 64-bit OS and is powered by a 2.4 GHz quad-core Arm Cortex-A76 processor. It incorporates a 64 GB SD Card for ample storage and features USB and HDMI ports to facilitate connectivity with peripheral devices, including our camera and touchscreen.
- **Touchscreen**: Has a 7-inch display and supports a hardware resolution of 1024 x 600 alongside a software resolution of 1920 x 1080. It is designed with HDMI and USB connections for integration with the Raspberry Pi.
- **Camera**: Has a 2 MP sensor and supports 1080p resolution for high-quality image capture. Equipped with a USB connection and capable of recording at 30 frames per second and is optimized for connection with a Raspberry Pi. This setup is utilized to monitor and analyze the user's posture, effectively distinguishing between correct and incorrect positioning.

IV. Standards

The project will have three crucial standards applied to ensure quality and security:

- **NIST Technical AI Standards [6, Th.1]** This standard states that the responsibility of programmers is to work across the government and with the industry to identify critical standards of development, strategies, and gaps in AI programming. Essentially, this means working cooperatively with other AI engineers to develop software that is useful to others in the AI community.
- **ACM Professional Responsibilities [5, Sec.2.5/2.6]** This standard mandates that computing professionals in a position of trust are responsible for outlining objectives and providing credible evaluations of their programs to the public. It is important to make the data and program open-source and remain completely open with consumers about how data is processed, who has access to it, and how it will be stored in relation to the AI model. Moreover, this standard requires that programmers only work in an area where they are competent, and applicable to the team with different levels of experience. Working alongside each other and absorbing information from professors and Ph.D. students will prove crucial to the successful implementation of the system.
- **NIST Technical AI Standards [6]** This standard mandates that all AI platforms adhere to international standards set in place to develop and deploy AI systems that help manage the many risks associated with AI programs and their data collection. It encourages the production of trustworthy and responsible AI systems. This is important for development as a host of data (personal pictures) gathered from many individuals will be used to train the AI model on posture. Ensuring the security of users is the number one priority.

V. Prototypes

A. Initial Sketches/Outlines

As exemplified in Figure 1, the pivotal function of our system revolves around the user's ability to "Select posture correction." The successful execution of this function necessitates the resolution of several core issues. After the user selects a posture correction type, the system initiates the process of detecting images or videos. This prompts the central query: How can ascertain whether the posture is correct or requires improvement?

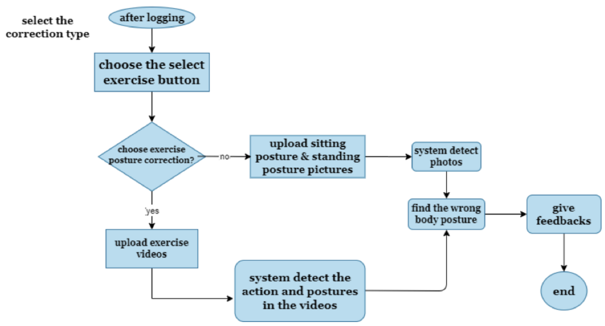

Fig. 1. Use case scenarios.

The initial plans for the front end were to have a main menu with options for selecting an exercise that would eventually open the camera to start getting feedback(as Figure 2 shows), an option to open a calendar to plan out exercises that are supported by the program and an option for setting or to login.

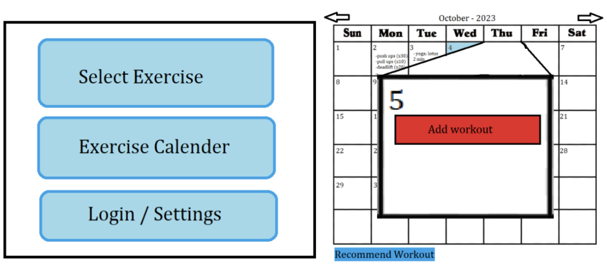

Fig. 2. Open exercise calendar, edit exercise calendar

B. First Prototypes

Upon the successful resolution of the aforementioned challenges, the subsequent focus shifts to the imperative issue of providing real-time feedback for the detection of incorrect postures [7]. Illustrated in Figure 3, this phase necessitates the delivery of user feedback aligned with predefined standards.

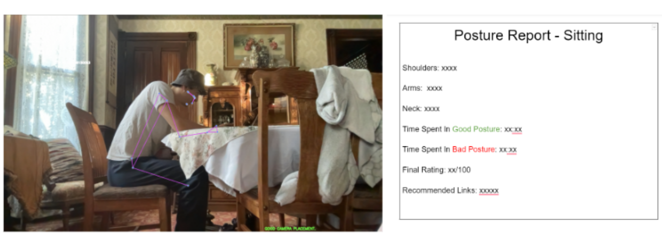

Fig. 3. Real-Time feedback

The underlying technology systematically scrutinizes the user's posture, assigning a "best of five" good/bad classification grading according to the quality and correctness of their stance with respect to the learned data. The users posture is recorded for 5 seconds (with a prediction made each second) to determine a score out of 5 for their posture. This sophisticated feedback mechanism not only enhances the user's awareness of their posture but also establishes a standardized metric for evaluating and improving their performance [8]. The integration of real-time feedback, coupled with a scoring system, represents a pivotal component of the system's functionality, further contributing to its overall effectiveness.

The prototype of the front end used PYQT5. As Figure 2 shows, the select exercise button did not have any functionality as the back-end was still being worked on. The calendar ended up supporting creating, editing, and deleting exercises as well as a recommended exercise button that you can choose to exercise your neck or spine from your upper body area which was the least used in the past week. The details of the exercise is shown in Figure 4.

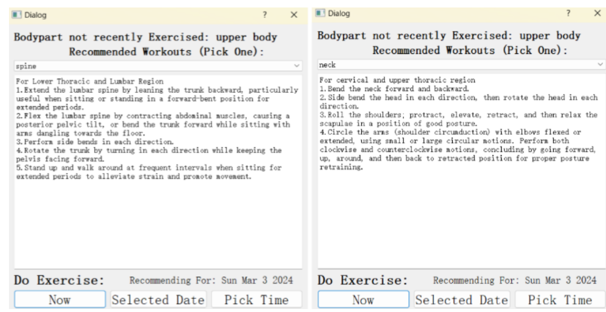

Fig. 4. Recommended exercise

The first machine learning model was trained via a straightforward approach, feeding the coordinates directly into the K-Nearest Neighbor (KNN) algorithm without any preprocessing. This yielded an 80% accuracy rate with it only improving up to an 83% accuracy rate with the implementation of a weighted KNN model.

C. Most Recent Prototypes

As of the end of the Winter Quarter 2024, the final prototype consists of 3 parts, the hardware, the front-end, and the back-end. The front-end is a PYQT5 application with a main menu, exercise calendar and camera GUI shown in Figures 5 and 6.

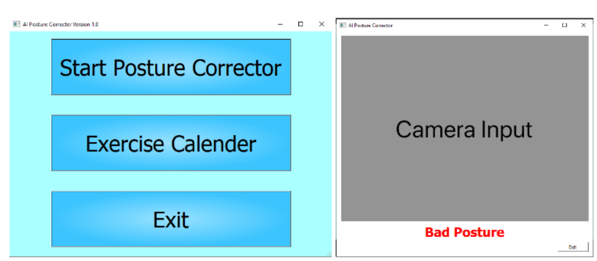

Fig. 5. Main Menu and Camera GUI

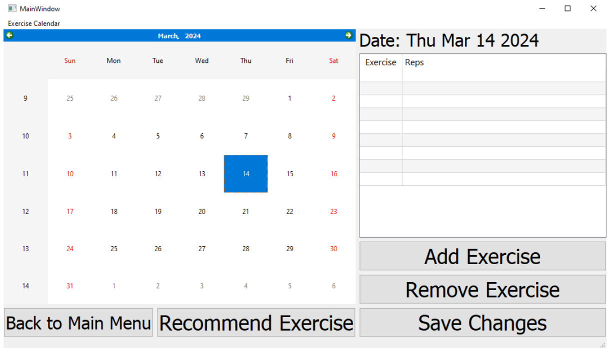

Fig. 6. Exercise Calendar

The current status of the back-end prototype involves the complete implementation of the OpenCV module to project estimated skeletons and extract data points in the form of 3D coordinates. The coordinates are then used as input to the Calculate Angle Formula shown in Figure 7. [9], [10].

```python
# Convert 'Good/Bad Posture' to numerical values
df['Good/Bad'] = df['Good/Bad'].map({'Good': 1, 'Bad': 0})

def calculate_angle(a, b, c):
    """
    Calculate the angle formed by three points.
    Points are provided as (x, y, z) tuples.
    """
    a = np.array(a) #First
    b = np.array(b) #Mid
    c = np.array(c) #End

    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)
    if angle >180.0:
      angle = 360-angle

    return angle
```

Fig. 7. Calculate The Angle Between Coordinates A, B, C

After calculating the angles for each data frame, data is fed into a Decision Tree (DT) AI Model to train the classifier. This classification model is designed to discern the distinctions between correct and incorrect postures to accurately predict the classification of incoming users' posture data [12]. Figure 8 shows the code used to set up the DT learner.

In the event that the model predicts a bad posture, it identifies specific areas of improvement and recommends corrective actions tailored to address the users posture related issues. For example, Figure 9 shows an output of our system in the event that a bad posture was detected due to the user excessively leaning forward. By considering articles from National Library of Medicine [13], the system advises the user on how they can minimize the risk of bad posture and back health.

To accumulate the back-end and front-end into a device of its own, the current prototype shown in Figure 10 marks our very first attempt at creating a hardware prototype. Consisting of a Raspberry Pi 5, Touchscreen, Power Supply, and a Camera Module, the AI Posture Correction System is ready to be put to practical use to predict the users posture, and provide constructive feedback to improve the users posture.

```python
#STORE THE VALUES OF ANGLES HERE
postureAngles = calculate_posture_angles(df)

# Adding the angle to the DataFrame
df['PostureAngle_FNH'] = postureAngles[0] #FNH
df['PostureAngle_NHK'] = postureAngles[1] #NHK

#new Dataframe with JUST 2 angles and Good/Bad posture
new_df = df[['Good/Bad', 'PostureAngle_FNH', 'PostureAngle_NHK']]

# Convert the DataFrame to a NumPy array
data_array_ang = new_df.to_numpy()

# Y is the first column
Y_ang = data_array_ang[:, 0]

# X is the rest of the columns
X_ang = data_array_ang[:, 1:]

##################################################################
#### MODEL TRAINED
##################################################################
model = DecisionTreeClassifier(max_depth=3)
model.fit(X_ang, Y_ang) #LOOK INTO JOBLIB TO SAVE MODEL
```

Fig. 8. Decision Tree - Model Setup

Leaning forward has resulted in bad posture. Here is how you can minimize the risk of bad posture and back health:
-avoiding sitting slumped to one side with the spine bent
-avoiding sitting for a long time in one position
-avoiding straining the neck for long periods while looking at a screen/document
For more info visit: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4499985/

Fig. 9. System's Posture Betterment Suggestion

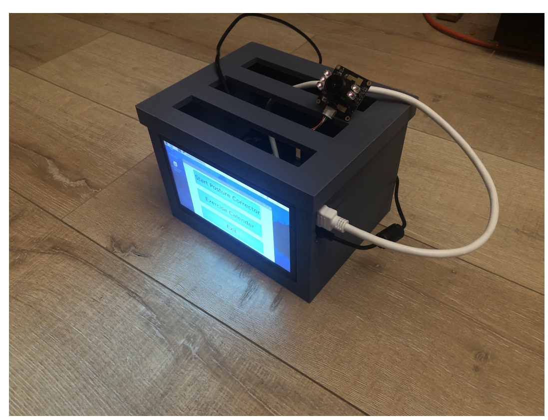

Fig. 10. Hardware Prototype

VI. Experiments/Results

A. Back-end Experiments/Results

The training dataset consists of 200 images depicting various seated postures, with an even number of images showcasing "good" and "bad" postures. These images were processed using OpenCV software to extract critical coordinates corresponding to the face, shoulders, and hips. The DT learner consists of hyperparameters that can be fine-tuned to adjust the learners' sensitivity to the data. The most significant hyperparameter within DTs is the "depth" of the tree. Figure 11 shows the comparison between learners' accuracy on the trained data as we change the depth hyperparameter.

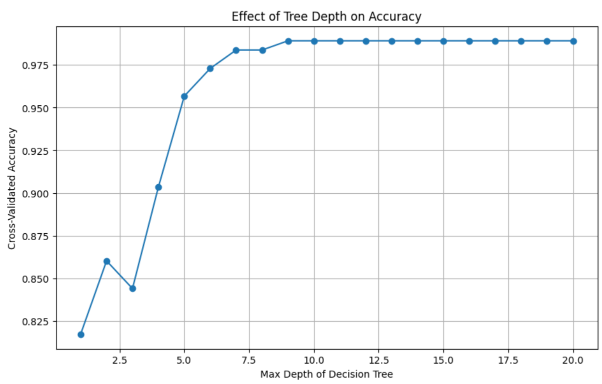

Fig. 11. Decision Tree - Depth Accuracy Comparison

Figure 11 suggests that models with a tree depth of 6 and above seem to overfit the data (i.e. may perform excellently on the given trained data but may predict new data inefficiently). By using a combination of depth 4 and 5, a model accuracy of 93% is achieved by the Decision Tree Classifier. A visualization of the model can be seen in Figure 12.

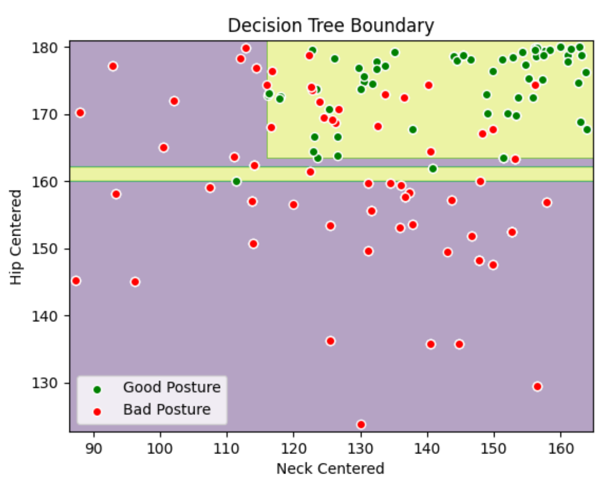

Fig. 12. Decision Tree - Model Visualization

Decision Trees offer a straightforward yet powerful approach for our AI Posture Correction System. Building upon these promising results, further improvements are envisioned for the model's capabilities. The next steps involve adding posture correction for various other activities such as weightlifting and yoga to further expand the use of this device. For the existing seated posture implementation, there is a planned expansion of the dataset to include a wider range of postures and body types, thereby enhancing the diversity and robustness of the training material. Additionally, exploration into more advanced machine learning algorithms and improvements to hardware (Camera) are planned, which could offer improved accuracy and efficiency.

B. Front-end Experiments/Results

For the front-end there are three major test cases: a good posture, a bad posture and an input with no detectable posture. In order to avoid potential false positives or negatives multiple frames are taken and processed for each update. Examples of a good and bad posture input are shown below in Figure 13.

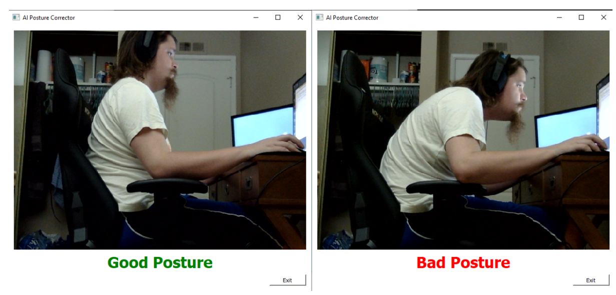

Fig. 13. Good/Bad Posture Test Cases

In the situation where some or all of the frames processed have no detectable posture, the message displayed with be "No Posture Detected". At least have of the processed frames must have a detectable posture to avoid this fail-safe. An example input for this case is shown below in Figure 14.

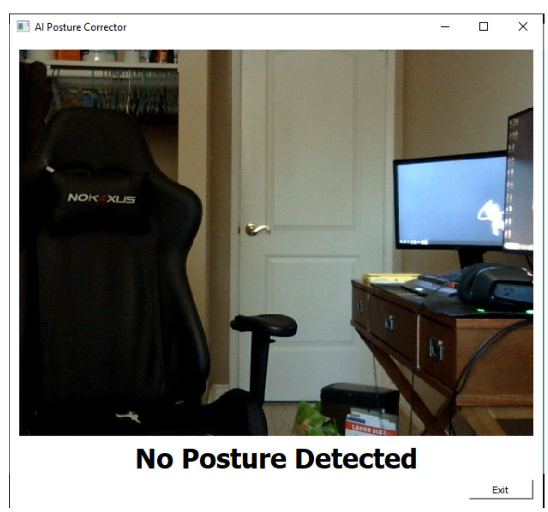

Fig. 14. No Posture Test Case

VII. Conclusions

The AI Posture Correction System aims to combat the posture epidemic by alerting and advising the users regarding their posture behaviors. Trained on Decision Tree machine learning model, the system uses its embedded camera to collect hundreds of data points from its users and preprocesses the data to feed the learner to predict an output. The users predicted posture estimation is then used to suggest improvements (i.e. exercises or altering sitting behaviors) [13]. Set to revolutionize ergonomic environments, this system bridges the gap between technology and health, promoting an integrated approach to posture correction and wellness.

References

[1] A. Syazwan et al., “Poor sitting posture and a heavy schoolbag as contributors to musculoskeletal pain in children: an ergonomic school education intervention program,” Journal of pain research, pp. 287–296, 2011.

[2] C. Z. YUE, L. C. YONG, and L. SHYAN, “EXERCISE QUALITY ANALYSIS USING AI MODEL AND COMPUTER VISION,” Journal of Engineering Science and Technology, pp.157–171, 2022.

[3] M. Palermo, J. M. Lopes, J. André, A. C. Matias, J. Cerqueira, and C. P. Santos, “A multi-camera and multimodal dataset for posture and gait analysis,” Scientific Data, vol. 9, no. 1, p. 603, Oct. 2022, doi: 10.1038/s41597-022-01722-7.

[4] Abhishek Kumar, Ebin Deni Raj, August 6, 2020, ”Silhouettes for Human Posture Recognition ”, IEEE Dataport, doi: https://dx.doi.org/10.21227/9c9b-3j44.

[5] Association for Computing Machinery, ”ACM Code of Ethics and Professional Conduct” acm.org, https://www.acm.org/code-of-ethics(accessed December 7, 2023).

[6] National Institute of Standards and Technology, ”Technical AI Standards” nist.gov, https://www.nist.gov/artificial-intelligence/technical-ai-standards(accessed December 7, 2023).

[7] P. V. Satish and M. Belwal, ‘Comparative Analysis of Machine Learning and Deep Learning Algorithms for Real-Time Posture Detection to Prevent Sciatica, Kyphosis, Lordosis’, in IoT Based Control Networks and Intelligent Systems, 2023, pp. 331–345.

[8] Chen, Steven, and Richard R. Yang. ”Pose Trainer: correcting exercise posture using pose estimation.” arXiv preprint arXiv:2006.11718 (2020).

[9] SARKER, I.H. “MACHINE LEARNING ALGORITHMS, REAL-WORLD APPLICATIONS AND RESEARCH DIRECTIONS,” SN COMPUT. SCI. 2, 160, 2021.

[10] K. Wakefield, “A guide to the types of machine learning algorithms and their applications,” SAS Inst. UK, 2021.

[11] O. Maimon and L. Rokach, Data Mining with Decision Trees: Theory and Applications. 2nd ed. World Scientific, 2014.

[12] N. Pulev, ”Understanding Decision Trees and Random Forests in Machine Learning,” 12 August 2022. [Online]. Available: https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees-and-random-forests-in-machine-learning/.

[13] Kim, D., Cho, M., Park, Y., Yang, Y., ”Effect of an exercise program for posture correction on musculoskeletal pain,” Journal of Physical Therapy Science, vol. 27, no. 6, pp. 1791-1794, 2015. [Online]. Available: https://doi.org/10.1589/jpts.27.1791