GUIDING ERGONOMIC HABITS WITH MACHINE LEARNING AND CAMERA-BASED ARTIFICIAL INTELLIGENCE FEEDBACK

Matthew Zhang [1], Carlos Gonzalez [2]

[1] Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Prolonged computers use fuels a growing epidemic of poor posture and related musculoskeletal issues, impacting quality of life and productivity. Addressing this, we propose a lightweight, real-time posture monitoring system designed for continuous background operation[1]. Utilizing Google's MediaPipe for pose detection and a heuristic-based scoring algorithm, our program analyzes key metrics like neck and torso angles [2]. The core challenge was objectively defining "good" vs. "bad" posture, which we addressed empirically with weighted metrics and an optimal threshold of 60.0. Experiments, using a 10,000-pose dataset, demonstrated 83.33% accuracy, with torso and neck angles proving most influential. This tool provides personalized end-of-day reports, leveraging AI (e.g., OpenAI's ChatCompletion API) to offer evidence-based recommendations[3]. Unlike specialized hardware or exercise-specific solutions, our camera-based application offers an accessible, continuous, and preventive approach for all prolonged computer users, fostering healthier digital habits.

KEYWORDS

Posture Monitoring, MediaPipe Pose Detection, Musculoskeletal Health, Heuristic Scoring Algorithm

1. INTRODUCTION

The problem I’m trying to solve is the growing issue of poor posture during prolonged computer use, especially among gamers and office workers. As digital lifestyles become more prevalent, people are spending extended hours sitting in front of screens with improper posture [4]. This has led to a rise in musculoskeletal problems such as chronic back pain, neck strain, and repetitive stress injuries.

Historically, concerns about posture were mostly confined to office ergonomics, but with the surge in recreational gaming and remote work, poor postural habits are now common across various age groups and settings. Many individuals are unaware of the damage caused by sustained poor posture until symptoms become chronic or debilitating. The lack of accessible tools to guide and correct posture in real time has only made the problem worse.

This issue is important because musculoskeletal disorders (MSDs) not only reduce quality of life but also place a significant burden on public health systems and workplace productivity [5]. Poor posture can lead to long-term injuries, increased medical costs, and reduced mobility and comfort

David C. Wyld et al. (Eds): CMLA, CSEIT, NLPD, GRAPH-HOC, SPM, NeTCoM, NCS, ArIT, DMSE – 2025
pp.141 -151, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151312

in daily life. If not addressed early, these issues can persist for years and contribute to other complications such as nerve compression or joint degeneration.
In the long run, this problem affects a wide population: gamers, students, remote workers, office employees, and anyone who uses a computer for extended periods. As more people shift toward digital-based work and entertainment, this health risk is expected to increase.

To address this issue, I developed a tool that promotes better posture awareness during daily activities. It incorporates real-time posture tracking, instructional feedback, and corrective exercises to encourage immediate and long-term improvements. This not only helps gamers improve their physical health but also benefits office workers, students, and anyone with sedentary screen time by preventing injury and promoting sustainable habits.

According to the U.S. Bureau of Labor Statistics, back-related musculoskeletal disorders accounted for approximately 38.5% of all work-related musculoskeletal disorders in 2016. Moreover, research shows that 42% to 69% of office workers report experiencing neck pain, and 31% to 51% report lower back pain (Cagnie et al. 68). These statistics highlight the scale of the issue and the urgency for effective preventive tools and interventions.

We explored three distinct methodologies related to posture and exercise monitoring. Arrowsmith et al. (2024) aimed to classify physiotherapy exercises and provide feedback for rehabilitation using single camera pose detection and machine learning. Its shortcoming was its focus on active, guided therapy, overlooking continuous, passive posture habits in office settings. Pereira et al. (2023) developed a mobile app for home physical therapy monitoring, leveraging Google MediaPipe with a high-precision Qualisys Motion Capture System for real-time feedback. Its primary limitation was the reliance on expensive, specialized hardware, making it impractical for daily use by most office workers. Lastly, Bourahmoune and Amagasa (2023) presented an AI-powered smart cushion (LifeChair) for real-time sitting posture recognition and stretching guidance using pressure sensors. While highly accurate, its drawback is the dependence on a specific hardware device, limiting accessibility and neglecting broader postural analysis.

Our project addresses these shortcomings by offering a lightweight, camera-based, heuristic-driven system that continuously monitors everyday sitting posture without requiring specialized equipment. We improve upon these works by providing holistic, end-of-day reports focused on long-term habit formation and prevention of musculoskeletal issues, offering a more accessible, ubiquitous, and non-intrusive solution for promoting ergonomic well-being in digital workplaces.

The tool offers a lightweight, real-time posture monitoring system that generates personalized reports to guide users toward improved habits and long-term musculoskeletal health. This approach solves the problem of neglected posture in digitized workplaces by providing continuous feedback without impacting computer performance.

The effectiveness of this solution stems from its unique combination of continuous monitoring and personalized reporting. Unlike methods that rely on static assessments or require conscious user input, the tool runs discreetly in the background, making it a seamless part of the user's workday. The real-time camera-based supervision allows for a comprehensive analysis of sitting patterns throughout the day, which is then summarized in an easy-to-understand report. This report, generated with the help of OpenAI's ChatCompletion API, not only highlights areas for improvement but also links directly to reliable studies and exercises. This integrated approach is significantly better than other methods because it addresses the core issue of habit formation and provides direct, actionable steps rather than just identifying problems.

Users can review their end-of-day reports to analyze their sitting patterns, understand where improvements can be made, and learn about specific exercises tailored to their needs.The implications for this tool are significant, especially as we continue to move towards a digitized workplace where a considerable amount of work is done on computer systems. In such environments, body posture and overall physical health can easily be neglected. This application aims to shed light on these commonplace yet critical issues, fostering a healthier and more sustainable working environment for individuals spending extended periods at their desks.

Our primary goal was to meticulously analyze the accuracy of our posture assessment model in distinguishing between good and bad sitting postures, which is crucial for its reliability. To test this, we designed an experiment utilizing a diverse subset of the "10,000 People - Human Pose Recognition Data" dataset, carefully balancing demographics and poses. The setup involved processing each image through our MediaPipe-based system, calculating neck angle, torso angle, and shoulder alignment, and comparing these metrics against ground truth labels.

The most significant finding was an overall accuracy of 83.33% at an optimal threshold of 60.0. Feature correlation analysis revealed that torso angle (0.583) and neck angle (0.460) were the strongest indicators of posture quality. We observed clear distinctions: good posture exhibited more consistent neck (mean: 164.51°) and torso angles (mean: 175.52°) compared to bad posture. The unexpectedly high accuracy at a lower threshold suggests our heuristic scoring system effectively captures posture quality. The torso angle emerged as the most influential factor, driving the most consistent results.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Server Optimization for Real-Time Feedback

A major component of our program is ensuring low latency for a responsive user experience. One key consideration is the server architecture used to deploy the application. For a real-time application, it’s important to consider and evaluate different protocols, such as HTTP versus TCP, and assess how they impact performance and responsiveness [6]. TCP could be chosen for its faster and more stable connection handling, particularly when integrated with frameworks like Flask and Gunicorn. One would also need to consider the deployment tools like WSGI to manage Python web apps effectively. Cost and scalability also factor in, as efficiency in both computational costs and response times are important in this context.

2.2. Adapting to Variable Camera Angles

Another crucial component of the program is ensuring accessibility regardless of how users position their cameras. We would need to consider various orientations—whether users are facing the camera directly, placing it beside them, using a phone angled from below, or a webcam above. To address these variables, one could design a generalized heuristic scoring system that evaluates posture based on normalized shoulder positions. Normalizing coordinate data would help standardize input, making the system more adaptable across different camera setups. This approach would ensure that the tool is inclusive and accessible for all users, without requiring ideal camera placement conditions, or different hardcoded scenarios, which would be too inefficient to design.

2.3. Defining and Personalizing Posture Scores

Another crucial challenge is accurately determining what constitutes a "good" versus "bad" posture score. Human posture can be subjective and vary significantly between individuals. One could address this by developing a multi-faceted scoring algorithm that considers various anatomical landmarks and their relative positions, rather than relying on a single point. To establish objective thresholds, one could leverage a dataset of diverse posture examples, some verified by ergonomic experts, to train a model or derive empirical boundaries. Furthermore, a personalized baseline could be integrated, allowing the system to learn a user's natural neutral posture and provide feedback relative to their own body, enhancing accuracy and user relevance.

3. SOLUTION

The system can broken down into three key components: the frontend interface, which was designed using the Flutter framework to provide an intuitive and interact experience for users on the web; the backend server, which listens for requests made from the web app to provide the scoring system as well as generative AI feedback; and the user data storage, which is set up using Google's NoSQL Firestore database to save historical user posture data.

The user is prompted to log into our services using the Google Authenticator. This is important since the system provides personalized feedback, which is determined by the unique UID provided from the Google Cloud Platform (GCP) [7]. Then, the user is redirected to the home page, where they can turn on their camera to begin assessment of their posture. The tool is designed so that the user can be sitting at their desk, and the posture-scoring system can then score the user in the background. The camera frame data gets sent to the backend, where the processes outlined from data extraction tools through MediaPipe and Python are performed. The web app then receives the final output, including various metadata, such as the final posture score, the angles of metrics such as the neck, shoulder, and head.

If the user wants a summarized overview of their posture over time, the web app also provides a tool where the user can receive an AI-generated summary outlining where the user can make improvements in the posture. We feed contextually-relevant user data to the LLM and prompt the system to provide feedback based on common patterns observed overtime.

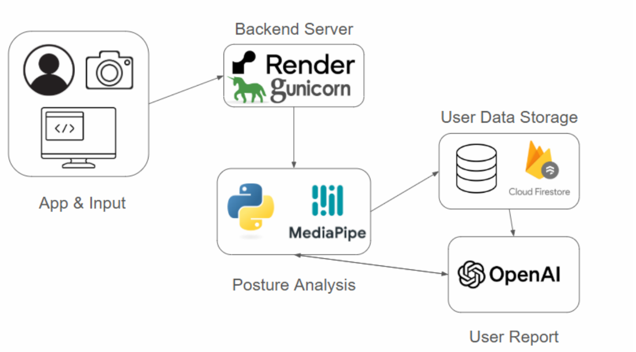


Figure 1. Overview of the solution

The Frontend Interface, built with Flutter, serves as the user's interactive gateway. Its purpose is to provide an intuitive web experience, allowing users to log in via Google Authentication (a security concept ensuring user identity), activate their camera for posture assessment, and view personalized feedback [8]. It communicates with the backend, sending camera data and receiving posture scores and AI summaries.

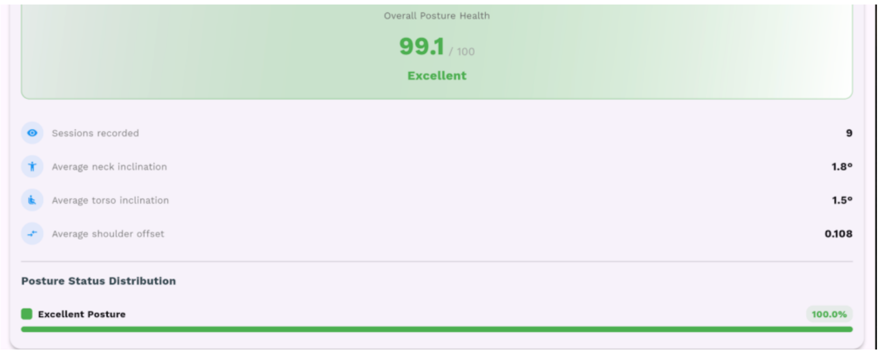


Figure 2. Screenshot of the interface

```
Future<void> _createAccount() async {
  try {
    final credential = await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: _emailController.text,
      password: _passwordController.text,
    );

    // Create initial user document in Firestore
    await FirebaseFirestore.instance
        .collection('users')
        .doc(credential.user!.uid)
        .set({
      'email': credential.user!.email,
      'nickname': _nicknameController.text,
      'createdAt': FieldValue.serverTimestamp(),
    });

    _showSuccessSnackbar('Account created successfully!');
  } on FirebaseAuthException catch (e) {
    if (e.code == 'weak-password') {
      _showErrorSnackbar('The password provided is too weak.');
    } else if (e.code == 'email-already-in-use') {
      _showErrorSnackbar('An account already exists for this email.');
    } else {
      _showErrorSnackbar('Error: ${e.message}');
    }
  } catch (e) {
    _showErrorSnackbar('Error creating account: $e');
  }
}
```

Figure 3. Screenshot of code 1

This code sample manages user account creation for a web application, which runs when a user submits a registration form to either sign up or log into our web app. The _createAccount() method attempts to create a new user. It first contacts Firebase Authentication (a backend server) using createUserWithEmailAndPassword(), sending the user's email and password. If successful, Firebase returns a UserCredential with a unique user ID (UID).

Next, the code interacts with Firebase Firestore (another backend server, a NoSQL database) via set(). It creates a document in the 'users' collection, using the UID as the document's identifier. This document stores personalized user data like the user's email, nickname, and a creation timestamp. _showSuccessSnackbar() and _showErrorSnackbar() provide front-end feedback, based on the outcome of the program. The backend servers, Firebase Authentication and Firestore, securely manage user accounts and store personalized data, with the UID serving as the primary user identifier. This becomes important in the future, when the user wants to collect a personalized user report based on their previous camera footage.

The Backend Server serves as the computational core of the system. Its purpose is to process real-time camera frame data from the frontend, analyze posture, and generate AI-driven feedback. The server receives camera data, extracts skeletal landmarks via MediaPipe, calculates posture scores and angles (e.g., neck, shoulder, head), and, for summarized overviews, feeds relevant user data to a Large Language Model (LLM) to provide contextualized improvement suggestions [9].

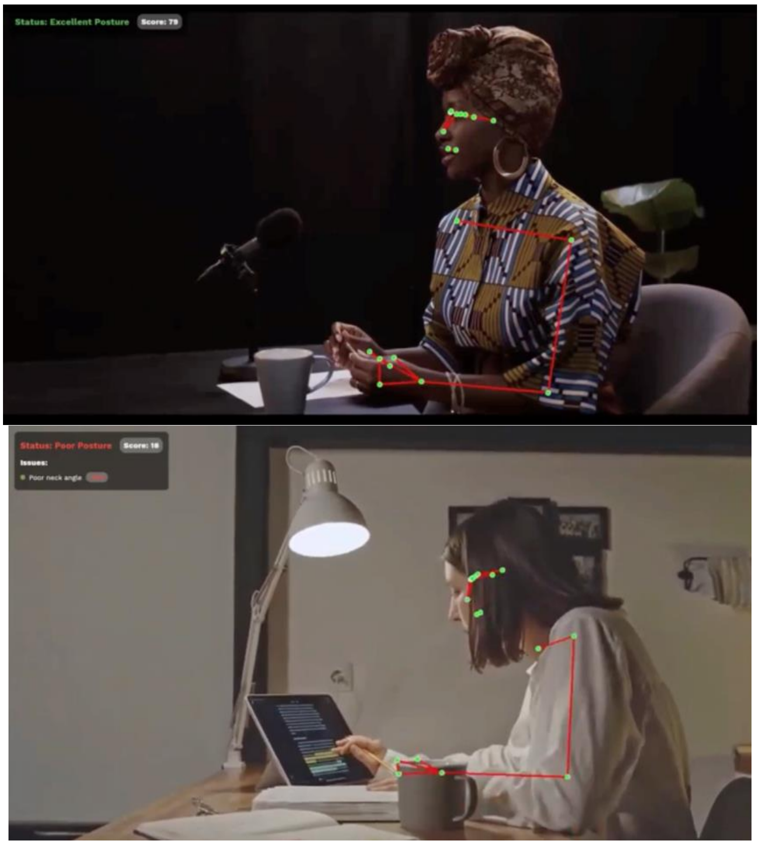


Figure 4. Screenshot of the body posture

The pseudocode for our posture-scoring mechanism is seen below:

```
FUNCTION calculatePostureScore(orientation, metrics):
    INITIALIZE score to 100 and empty issues list.
    CALCULATE angles (neck, torso) and joint differences (shoulders,
head).
    APPLY initial score penalties and add issues for poor neck angle.
    SCORE shoulder alignment based on `orientation` and add issues.
    SCORE neck, torso, and head position components, adding issues
for deviations.
    AGGREGATE component scores to derive `total_score`.
    CLAMP `total_score` between 0 and 100.
    RETURN `total_score` and `issues`.

FUNCTION draw_landmarks_efficiently(image, results):
    IF no `results.pose_landmarks`: RETURN `image`.
    CREATE `overlay` from `image`.
    DRAW defined `connections` between key landmarks on `overlay`
with specific color/thickness.
    DRAW key `landmarks` as circles on `overlay` with specific
color/size.
    BLEND `overlay` with original `image` for final output.
    RETURN blended `image`.
```

Figure 5. Screenshot of code 2

This Python code, running on the backend server, processes real-time posture data. It executes continuously when a user's camera is active and transmitting frames for posture assessment.
The calculatePostureScore method determines a numerical posture score (0-100) and identifies specific posture issues. It calculates angles (e.g., neck, torso) and distances (e.g., shoulder alignment) from the metrics (landmark coordinates received from the frontend). It then applies penalties and assigns scores based on these calculations, accumulating 'issues' like "Poor neck angle" with a severity. The final score aggregates these individual component scores.

The draw_landmarks_efficiently method takes an image frame and the processed results (containing pose landmarks). It visually overlays key body connections and landmarks (shoulders, hips, ears, nose) onto the image, enhancing the visual feedback.

Variables like base_score, issues, neck_angle, shoulder_score, etc., are created within the calculatePostureScore function to store intermediate calculations and the final output. On the backend, this server receives camera data, performs MediaPipe processing to extract metrics, and then uses this code to evaluate posture, sending the score and visual feedback back to the frontend.

The User Data Storage component, implemented with Google Cloud Firestore, serves to persistently save historical user posture data [10]. Its purpose is to provide personalized feedback and enable trend analysis.This component functions by storing raw posture metrics and derived summary statistics into a Firestore NoSQL database, which are then used by the backend for generating reports and contextualizing AI feedback.

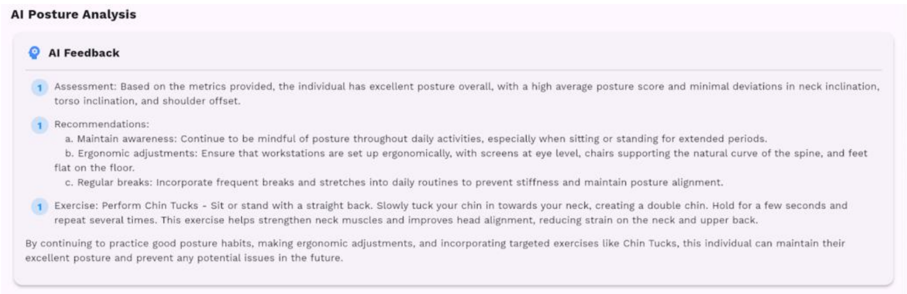


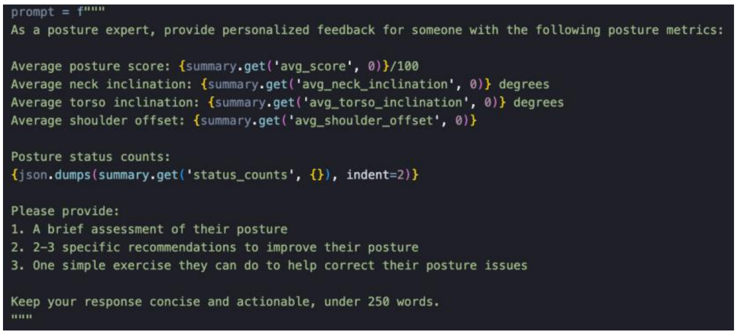
Figure 6. Screenshot of AI analysis

The pseudocode for the OpenAI chatcompletion request is seen below:

```
FUNCTION generate_posture_feedback():
    IF OpenAI client not available: RETURN error.
    GET 'summary' data from request.
    IF no summary data: RETURN error.

    CONSTRUCT `prompt` using summary data.
    CALL OpenAI API with `prompt` and model parameters.
    EXTRACT feedback content from API response.
    RETURN success with `feedback`.

    ON API_ERROR: RETURN error message.
    ON GENERAL_ERROR: RETURN error message.
```



```
prompt = f"""
As a posture expert, provide personalized feedback for someone with the following posture metrics:

Average posture score: {summary.get('avg_score', 0)}/100
Average neck inclination: {summary.get('avg_neck_inclination', 0)} degrees
Average torso inclination: {summary.get('avg_torso_inclination', 0)} degrees
Average shoulder offset: {summary.get('avg_shoulder_offset', 0)}

Posture status counts:
{json.dumps(summary.get('status_counts', {}), indent=2)}

Please provide:
1. A brief assessment of their posture
2. 2-3 specific recommendations to improve their posture
3. One simple exercise they can do to help correct their posture issues

Keep your response concise and actionable, under 250 words.
"""
```

Figure 7. Screenshot of code 3

This Python code defines a backend endpoint, generate_posture_feedback, that runs when the frontend requests AI-generated posture feedback. This occurs when the user visits the report page, and a report gets generated based on the timeframe the user requests the data from.

The method first validates that the OpenAI client is available and that summary data (containing posture metrics) is provided in the request. It then dynamically constructs a textual prompt by embedding the user's summary data into a predefined template. This prompt acts as instructions for the LLM.

The core of the method is the call to the OpenAI API. It sends the prompt to the gpt-3.5-turbo model. The OpenAI server processes this prompt and generates a personalized response based on the provided posture metrics. The server then sends this generated feedback back to the backend. Finally, the backend extracts this feedback and returns it to the frontend for display to the user. Variables like data, summary, prompt, completion, and feedback_content are used to manage the request, construct the AI query, and handle the response. We fallback on various error cases if situations arise, in cases like a lack of data (e.g.: the user hasn’t used the camera feature, and wants to generate a report based on no previous data).

4. Experiment

We want to analyze the accuracy of our model in distinguishing between good and bad posture. Thus, we aim to design an experiment which can explicitly test for these scenarios, where we use a sample dataset to help in our model evaluation.

The experiment was designed to evaluate the accuracy of our posture assessment system using a diverse dataset of 10,000 human pose images. We selected a balanced subset of images representing various demographics, environments, and poses to ensure comprehensive testing. The experiment processes each image through our MediaPipe-based pose detection system, calculating three key metrics: neck angle, torso angle, and shoulder alignment. These metrics are weighted based on their correlation with good posture (40%, 35%, and 25% respectively). The system's predictions are compared against ground truth labels to evaluate accuracy. We sourced our control data from the "10,000 People - Human Pose Recognition Data" dataset, which provides a diverse and well-annotated collection of human poses across different demographics and conditions.

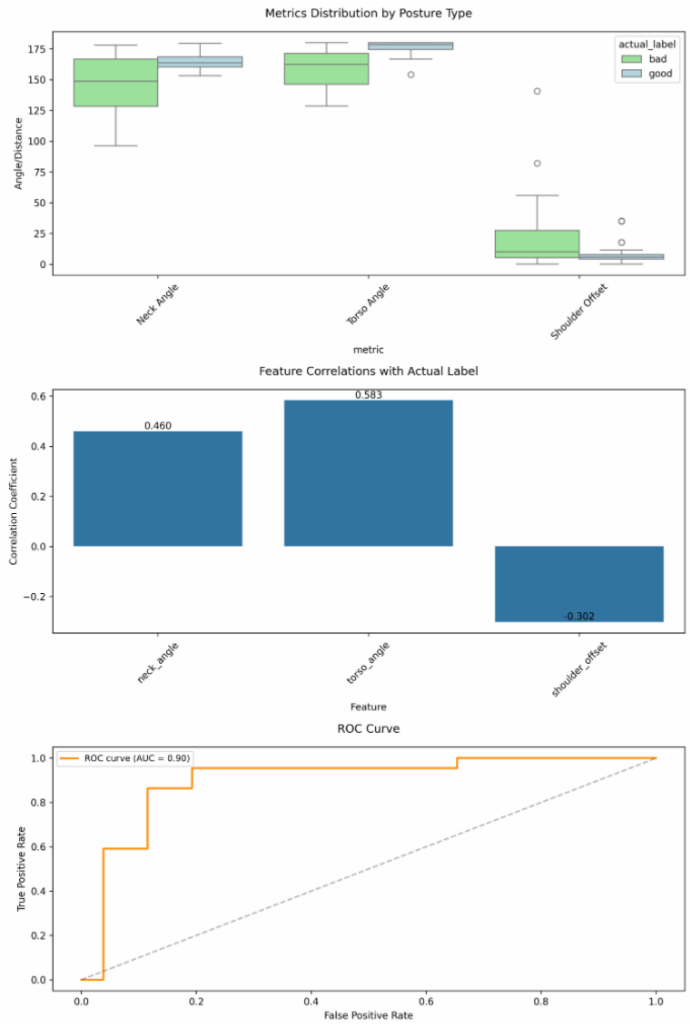


Figure 8. Figure of experiment

=== Threshold Analysis ===
Threshold: 60.0
Accuracy: 83.33%
False Negatives: 1.0
False Positives: 7.0
===Feature Correlation===
Neck Angle: 0.4597904888636265
Torso Angle: 0.5830636305058006
Shoulder Offset: -0.3017384850784524

The experiment yielded several key insights. The model achieved an overall accuracy of 83.33% at the optimal threshold of 60.0, with only 1 false negative and 7 false positives. The feature correlations revealed that torso angle (0.583) and neck angle (0.460) were the strongest predictors of posture quality, while shoulder offset showed a negative correlation (-0.302). The data showed clear separation between good and bad posture: good posture samples had more consistent neck angles (mean: 164.51°, std: 6.52°) compared to bad posture (mean: 146.33°, std: 23.49°). Similarly, good posture demonstrated more stable torso angles (mean: 175.52°, std: 6.48°) versus bad posture (mean: 158.14°, std: 15.63°). The most surprising finding was the high accuracy achieved with a relatively low threshold (60.0), suggesting that our scoring system effectively captures posture quality even with more lenient criteria. The biggest effect on results came from the torso angle measurement, which showed the strongest correlation with posture quality and the most consistent measurements across good posture samples.

5. RELATED WORK

A scholarly source tackling a related problem is Arrowsmith et al. (2024), which focuses on physiotherapy exercise classification using single-camera pose detection and machine learning [11]. Their solution involves training CNN and SVM models to classify different exercises and provide feedback on body movement, adjustments, and alternative exercises to ensure adherence to best practices for rehabilitation. While effective for classifying discrete exercises and guiding specific movements, its primary limitation is its focus on active, guided physical therapy rather than the passive, sustained posture of an office worker. It largely ignores the nuances of subtle postural deviations over an entire workday or the development of long-term sitting habits. Our project improves upon this by specifically targeting continuous, background monitoring of everyday sitting posture using a lightweight, heuristic-based system, providing holistic end-of-day reports focused on habit formation and prevention in a non-clinical context.

Another relevant scholarly work is Pereira et al. (2023), who developed a machine learning app for monitoring physical therapy at home [12]. Their solution utilizes Google MediaPipe to collect landmark data, which is then fed into a high-precision Qualisys Motion Capture System to provide real-time feedback on user exercise performance. This method offers high precision for specific exercise monitoring and real-time guidance. However, its significant limitation for our problem is the reliance on a specialized, high-cost Qualisys system, which is impractical for widespread everyday use by office workers. It also focuses solely on active exercise feedback rather than continuous passive posture. Our project offers a superior solution by being entirely camera-based and heuristic-driven, eliminating the need for expensive motion capture hardware. This allows our tool to run discreetly in the background, offering an accessible and lightweight application that monitors continuous sitting patterns and generates personalized, actionable end-of-day reports for long-term posture improvement and well-being.

Bourahmoune and Amagasa (2023) present a different approach, using the LifeChair smart cushion, which combines pressure sensing technology, a smartphone app, and machine learning for real-time sitting posture recognition and seated stretching guidance [13]. This system achieves a high accuracy (98.93%) in detecting various sitting postures by considering user BMI. While this is effective in identifying different postures and guiding stretching, it relies on a specific hardware device (the smart cushion), limiting its accessibility and potentially its ability to capture subtle postural changes that a camera could detect. It also focuses on posture recognition and correction through stretching, rather than continuous monitoring and habit formation. Our system, by using a camera, offers a more ubiquitous and less intrusive solution, focusing on long-term habit analysis and providing personalized reports to improve overall sitting patterns without requiring additional hardware.

6. CONCLUSIONS

One significant limitation of our current project is its reliance on users having access to a consistent camera setup on their primary computing device. To enhance accessibility and flexibility, future iterations could integrate a mobile application feature, allowing users to leverage their phone cameras for posture monitoring, which would be particularly useful for office workers who might not have webcams or prefer mobile-based solutions [14]. Furthermore, while our heuristic-based scoring system is effective, it could benefit from continuous refinement. With more development time, we would implement an adaptive learning mechanism to adjust model parameters based on individual user sitting habits over time. This personalization would allow the system to more accurately define "good" and "bad" posture uniquely for each user, moving beyond a generalized approach and ensuring the feedback provided is even more relevant and actionable for long-term ergonomic health.

Our tool makes significant strides toward improving digital well-being, leveraging state-of-the-art technologies for precise data collection and robust inference [15]. By integrating LLM and AI technology, we can better inform individuals about their common sitting habits, ultimately guiding them toward healthier, more sustainable work practices in our increasingly digitized world.

REFERENCES

[1] Anwary, Arif Reza, et al. "Smart-Cover: A real time sitting posture monitoring system." Sensors and Actuators A: Physical 317 (2021): 112451.

[2] Amprimo, Gianluca, et al. "Hand tracking for clinical applications: validation of the Google MediaPipe Hand (GMH) and the depth-enhanced GMH-D frameworks." Biomedical Signal Processing and Control 96 (2024): 106508.

[3] Rawashdeh, Ahmad, Omar Rawashdeh, and Mohammad Rawashdeh. "ChatGPT and ChatGPT API: An Experiment with Evaluating ChatGPT Answers." Proceedings of the Future Technologies Conference. Cham: Springer Nature Switzerland, 2024.

[4] Kushwaha, Bijay Prasad. "Paradigm shift in traditional lifestyle to digital lifestyle in Gen Z: a conception of consumer behaviour in the virtual business world." International Journal of Web Based Communities 17.4 (2021): 305-320.

[5] National Research Council, et al. "Musculoskeletal disorders and the workplace: low back and upper extremities." (2001).

[6] Charzinski, Joachim. "HTTP/TCP connection and flow characteristics." Performance Evaluation 42.2-3 (2000): 149-162.

[7] Bisong, Ekaba. "An overview of google cloud platform services." Building Machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners (2019): 7-10.

[8] Wijayarathna, Chamila, and Nalin AG Arachchilage. "An empirical usability analysis of the google authentication api." Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering. 2019.

[9] Kasneci, Enkelejda, et al. "ChatGPT for good? On opportunities and challenges of large language models for education." Learning and individual differences 103 (2023): 102274.

[10] Varshney, Heena, Ali S. Allahloh, and Mohammad Sarfraz. "Iot based ehealth management system using arduino and google cloud firestore." 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON). IEEE, 2019.

[11] Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." Communications of the ACM 56.1 (2013): 116-124.

[12] Akkarakittichoke, Nipaporn, et al. "Characteristics of office workers who benefit most from interventions for preventing neck and low back pain: a moderation analysis." Pain reports 7.3 (2022): e1014.

[13] Arrowsmith, Colin, et al. "Physiotherapy exercise classification with single-camera pose detection and machine learning." Sensors 23.1 (2022): 363.

[14] Pereira, Bruno, et al. "A machine learning app for monitoring physical therapy at home." Sensors 24.1 (2023): 158.

[15] Bourahmoune, Katia, and Toshiyuki Amagasa. "AI-powered Posture Training: Application of Machine Learning in Sitting Posture Recognition Using the LifeChair Smart Cushion." IJCAI. 2019.