A MOBILE CAMERA SYSTEM TO ASSIST IN MAINTAINING BETTER POSTURE THROUGH THE USE OF COMPUTER VISION AND ARTIFICIAL INTELLIGENCE

Abby Zhang[1], Andrew Park[2]

[1]Deep Run High School, 4801 Twin Hickory Rd, Glen Allen, VA 23059
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In an increasingly digital world, maintaining good posture is essential for overall health and well-being. This paper introduces a novel solution, a mobile camera system leveraging computer vision and artificial intelligence (AI) technologies, to assist individuals in maintaining better posture. The system utilizes the camera on a mobile device to continuously monitor the user's posture in real-time. Computer vision algorithms analyze the user's body position and alignment, while AI algorithms provide personalized feedback and guidance based on established ergonomic principles. Through the integration of advanced technologies, the system aims to promote awareness of posture habits and encourage corrective actions to prevent musculoskeletal issues associated with poor posture. This paper discusses the design, implementation, and evaluation of the mobile camera system, highlighting its potential to revolutionize posture management practices and improve overall health outcomes in the digital age.

KEYWORDS

Posture, Artificial Intelligence, Raspberry Pi, Flutter

1. INTRODUCTION

Maintaining a strong and upright posture serves a key rolein the long-term health of people by reducing the risk of muscle tension, minimizing fatigue, enhancing the physical physique, and creating a better mood. However, many students are known to slouch when working at their desks for hours on end since it is an easier position to relax in. Statistics have reported to show that out of around 600,000 students, 65% of children and adolescents were deemed to show incorrect posture, and around 3.7% were referred to radiology [1]. Moreover, students will potentially worsen their posture over time, leading to severe damage to the spinal cord, which then results in several joint and degenerative diseases such as multiple sclerosis, arthritis, Parkinson’s disease, etc. Other studies have shown that posture worsens as age increases [2]. Additionally, as it’s evident that the generation today is already at risk, this problem may affect future generations in the long run as our technology becomes more advanced and prevalent, and students gradually increase their intake of computer usage throughout the years. Previous studies support this by showing that students have a direct correlation between the number of musculoskeletal problems with the number of hours on digital devices. One study revealed that 70% of students aged 10–12 years reported musculoskeletal symptoms in at least one part of their

David C. Wyld et al. (Eds): SE, SAIM, SIPM, CoNeCo, ICITE, ACSIT, CMIT, FCST, SNLP – 2024
pp. 103-113, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140810

bodies that are tied back to their usage of computers [3]. Therefore, a solution that can improve various people's posture to ensure that their overall health will be stable and healthy for decades will be discussed in the following reports.

Lauren Simpson, Monish M. Maharaj, and Ralph J Mobbs assessed the effects of wearables, technological devices worn on the body, on posture analysis. However, there are several shortcomings that are prevalent in this study. This includes the low accuracy of the device; validity, where they only tested on a small sample size; and practicality, where multiple sensors are required to be attached to the user.

Quilong Wan et. al goes a different direction through the use of a pressure sensor paired with a hip positioning system for analyzing sitting posture. Their system manages to reach a relatively high degree of accuracy at 89.6%. However, the major shortcoming comes in the form of a custom sensor array, which can prove to be both expensive and difficult to obtain.

The approach taken by Change et. al is focused on designing products in general for posture correction and even accounting for the consequences of bad posture such as myopia. Their prototype involves a Lego EV3 model that the user attaches to their body so that data can be collected, and notifications can be sent to them. With that being said, this has a similar drawback to the method done by Simpon et.al in that it requires physical strapping of devices, which can prove inconvenient.

Our project aims to improve the methods attempted by others by developing a computer vision-based system that remains unobtrusive and doesn't require you to wear physical attachments or custom sensors, providing a more accessible and friendly approach to posture correction.

To solve this problem, we proposed to use a device connected to a camera to analyze posture. This proposal solves the problem of bad posture because it alerts users to straighten their spines whenever they are seen slouching. To do so, we incorporate AI into a camera system that allows it to identify the slope of the back to determine whether or not one is slouching. Users will have a mobile app that can keep track of their statistics and notify them to straighten their back. This is an effective solution because it uses persistent monitoring that tracks the user's movement throughout the time it is turned on. This allows for a long period of working with good posture and is effective in the long term. Furthermore, the statistics and data contained in each recording enable users to identify their trends and progress. Due to this, they can make adjustmentson their own accord if necessary. Ultimately, this proposal is an innovative and efficient way to improve one's posture and health in the long run.

2. CHALLENGES

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

2.1. The Communication

One major issue that is necessary to consider is the communication between the Raspberry Pi and Flutter App. First of all, live video transmission between these two components requires low latency to reduce a less responsive and laggy experience for the user when viewing themselves in the app. To resolve this, I could use a combination of optimized video compression algorithms and efficient streaming protocols, ensuring a smooth and responsive live video feed. Additionally, integrating Firebase as the backend provides a reliable and scalable solution for data storage, retrieval, and synchronization between the Raspberry Pi and Flutter App. This not only facilitates real-time communication but also allows for seamless updates and data consistency. Ensuring robust Wi-Fi connectivity and establishing effective server communication protocols further contribute to the overall reliability and speed of data transfer between the Raspberry Pi and the Flutter App, creating a seamless and responsive user experience.

2.2. The AI Analysis Speed

Another component that posed a problem is the AI analysis speed. The challenge lies in ensuring a swift analysis to prevent the video feed from becoming excessively choppy and laggy. To address this concern, strategic steps such as preprocessing the image, resizing it to a smaller dimension, and employing image cropping techniques become essential. By implementing these measures, the image size is effectively compressed, facilitating a smoother execution of both image processing and video analysis. This optimization is particularly crucial for scenarios involving real-time video feeds, where sluggish AI analysis could significantly impact the seamless flow of information and compromise the effectiveness of the entire system.

2.3. The AI

Another major obstacle that we need to consider is that the AI is written in Python, but the mobile app is written using Dart. This poses an issue because the two languages cannot directly work together; we need some way for the video feed from the Flutter app to be transferred and analyzed by the AI before we show it to the end user. One potential means to achieve this would be to create a Flask server where the AI will reside while the Flutter app simply records the video feed and sends the raw video data to the server. The use of raw video data would mean there is no need for direct interoperability which circumvents the issue presented. It is important to mention that this proposal has the potential issue of latency, as the need to send data to a server for processing and waiting for a response will inherently add extra overhead which causes a laggier-looking result.

3. SOLUTION

The three main components making up the experience for PostureGuide is the mobile app, an optional Raspberry Pi, and the analysis pipeline. The mobile app allows access for users to monitor themselves through the Raspberry Pi's camera module. The AI analysis process is what measures and calculates the user's posture to determine whether or not they need to engage in better posture. This is connected to the Raspberry Pi camera module, which is the device that will record and view the user. The intended flow of the user is to create an account in our app, then choose their preferred method for monitoring to set up, which is using their phone camera or the device camera, and then learn from the information our application provides to determine any fixes in their posture. This program uses Flutter to develop the app, Python for the Raspberry Pi, and Mediapipe as the analysis component.

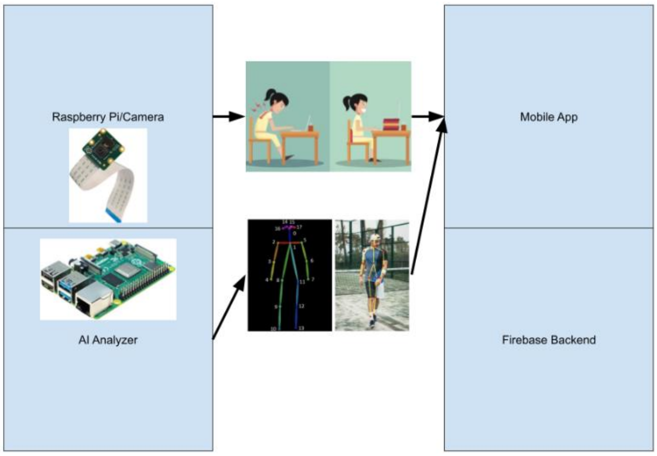

Figure 1. Overview of the solution

The Raspberry Pi is the core computer powering the device prototype that accompanies the app which monitors the user before analyzing the feed and notifying the user if they have bad posture! If the user elects to use this device, it will handle setup and connection completely headless.

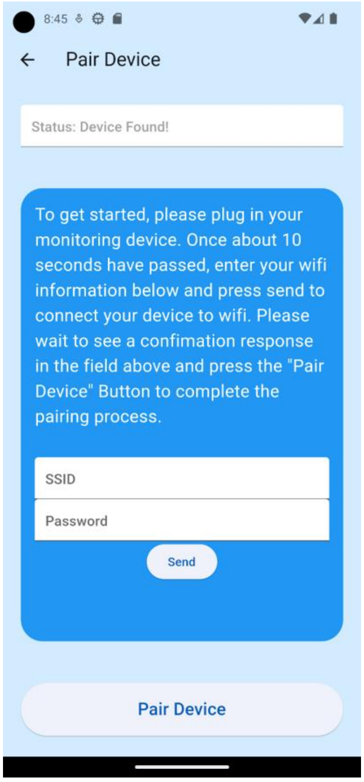

Figure 2. Screenshot of device function

```python
class WiFiCharacteristic(Characteristic):
    def __init__(self, bus, index, service):
        Characteristic.__init__(self, bus, index, WIFI_RX_CHARACTERISTIC_UUID,
                                ['write', 'read', 'notify'], service)
        self.notifying = True

    def WriteValue(self, value, options):
        received_data = ''.join([chr(byte) for byte in value])
        if ":" in received_data:
            ssid, password = received_data.split(":")
            self._value = connect_to_wifi(ssid, password).encode('utf-8')
        elif received_data == "get_ip":
            ip_address = get_ip_address()
            self._value = ip_address.encode('utf-8')
        elif received_data == "get_wifi_status":
            wifi_status = get_wifi_status()
            self._value = wifi_status.encode('utf-8')
        self.send_notification(self._value)

    def ReadValue(self, options):
        return self._value if self._value else b''

    def StartNotify(self):
        if self.notifying:
            return
        self.notifying = True

    def StopNotify(self):
        if not self.notifying:
            return
        self.notifying = False

    def send_notification(self, value):
        if not self.notifying:
            return
        dbus_value = [dbus.Byte(b) for b in value]
        print(f"Converted value: {dbus_value}")
        self.PropertiesChanged(GATT_CHRC_IFACE, {'Value': dbus_value}, [])
```

Figure 3. Screenshot of code 1

The WifiCharacteristic class is part of a large codebase that lives within the Raspberry Pi itself which allows it to be remotely given the wifi information so that it can connect to a new network. This is necessary as the video feed is streamed to an http endpoint which the device can only do if it is connected to the local network for other devices to view. If the Raspberry Pi determines that it is not connected to wifi, it spins up a Bluetooth server that our phone can connect to and provides the wifi information from the mobile app itself. Once this information is retrieved, the Bluetooth server will attempt to connect to the wifi network and respond with the IP address so that a connection can be established.

The Flutter App is the software that this project utilizes to create the app on both Android and iOS platforms with a single codebase. This will allow users to access and monitor themselves through their device, with their history or previous videos and posture trends saved.

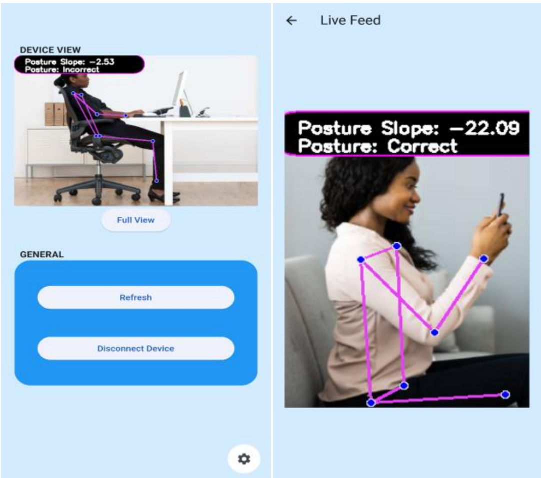

Figure 4. Screenshot of posture analysis

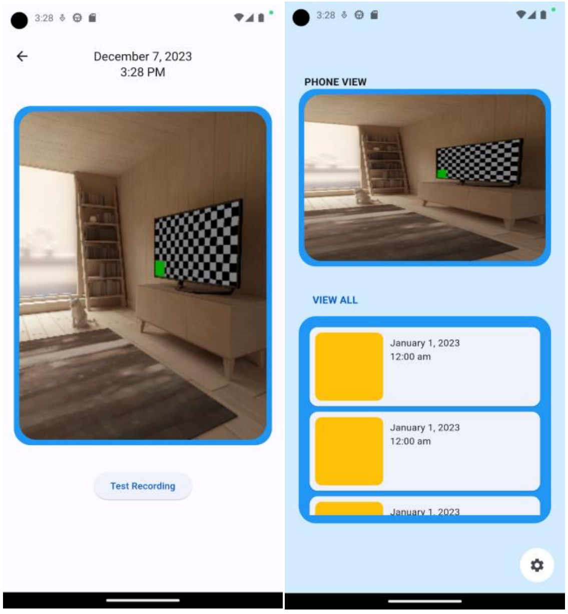

Figure 5. Screenshot of recording

```
import 'package:flutter/material.dart';
import 'package:my_app/size_config.dart';
import 'package:webview_flutter/webview_flutter.dart';
import 'package:my_app/shared/singleton.dart';

final _singleton = Singleton();
String rasp_ip = _singleton.rasp_ip;

class LiveFeed extends StatelessWidget {
  LiveFeed({super.key});

  final WebViewController controller = WebViewController()
    ..setJavaScriptMode(JavaScriptMode.unrestricted)
    ..setBackgroundColor(const Color(0x00000000))
    ..setNavigationDelegate(
      NavigationDelegate(
        onProgress: (int progress) {
          // Update loading bar.
        },
        onPageStarted: (String url) {},
        onPageFinished: (String url) {},
        onWebResourceError: (WebResourceError error) {},
        onNavigationRequest: (NavigationRequest request) {
          if (request.url.startsWith("https://www.youtube.com/")) {
            return NavigationDecision.prevent;
          }
          return NavigationDecision.navigate;
        },
      ),
    )
    ..loadRequest(Uri.parse("http://$rasp_ip:5000/video_feed"));

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      appBar: AppBar(
        title: const Text('Live Feed'),
        backgroundColor: const Color.fromARGB(255, 210, 235, 255),
      ), // AppBar
      body: Container(
        color: const Color.fromARGB(255, 210, 235, 255),
        width: SizeConfig.blockSizeHorizontal! * 100,
        height: SizeConfig.blockSizeVertical! * 100,
        child: Center(
```

Figure 6. Screenshot of code 2

This part of the application showcases some of the communication methods used to allow for a video feed from the Raspberry Pi to be viewed from within the app itself. Essentially a rasp_ip that was determined when setting up the device is used to allow the app to connect to an http endpoint that allows us to view the video stream within a widget.

The analysis process itself is the third major component of the project. Any video feed captured either by the phone or the device is sent to an AI for pose estimation so that we cant determine if the user is currently using good posture.

Figure 7. Screenshot of posture

```python
def get_pose(self, image):
    image.flags.writeable = False # improve performance
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = self._pose.process(image)

    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    return image, results


def _draw_landmark(self, result, frame):
    relevant_landmarks = {25, 26, 27, 28, 12, 11, 24, 23}

    for idx, landmark in enumerate(self._landmarks):
        if idx not in relevant_landmarks:
            self._landmarks[idx].visibility = 0

    self._mpDraw.draw_landmarks(frame, result.pose_landmarks, self._mpPose.POSE_CONNECTIONS,
        self._mpDraw.DrawingSpec(color=(255, 0, 0), thickness=2, circle_radius=2),
        self._mpDraw.DrawingSpec(color=(0, 255, 0), thickness=2, circle_radius=2),)

    return frame

def _fetch_posture_slope(self, image, h, w):
    left_shoulder = self._mpPose.PoseLandmark.LEFT_SHOULDER.value
    left_hip = self._mpPose.PoseLandmark.LEFT_HIP.value
    # mx + b

    m = slope(self._landmarks[left_hip].x * w, self._landmarks[left_hip].y * h,
              self._landmarks[left_shoulder].x * w, self._landmarks[left_shoulder].y * h)

    posture = "Correct"
    if -5 <= m <= 5: # adjust as needed if not accurate enough
        posture = "Incorrect"

    image = self._write_slope(m, image, posture)

    return image
```

Figure 8. Screenshot of code 3

The analysis process works by taking in a video feed and then feeding that information into the get_pose() function where the loaded model can analyze the frame to get pose information. The _draw_landmark() and _fetch_posture_slope() functions in turn visualize the results and also determine if the user is exhibiting good or bad posture by checking the various angles between landmarks of interest like the hips, knees, back, and shoulders.

4. EXPERIMENT

In order todetect blind spots in the project, a few experiments were conducted.

4.1. Experiment 1

One potential blind spot in my program that could be tested is the accuracy measurement contingent upon several environmental factors. This is a crucial aspect to my program because it is the main function that users would rely on to correct their posture when working.

There are X total factors that we will be seeking to investigate: lighting, camera angle, camera distance, subject's height, and general obstructions. To test each of the various factors and their impact on the overall system, we will isolate each of them in their batches for tests. For each batch, the control runs will be the user operating the system under ideal conditions with both good and bad posture while the experimental runs will be applying a variation relevant to the factor being tested. Ideal conditions can be defined as generally good lighting with the camera directly facing the subject with all of the subjects visible without any obstructions.

Here are the variations that will be done for each factor:

- Lighting: dimmer (X lumens) and brighter (X lumens) light settings
- Camera Angle: 45 degrees from the front and back, both sides
- Camera Distance: camera is closer, farther
- Subject Height: good and bad posture from all participants along with recorded height
- Obstructions: particle cover, complete cover

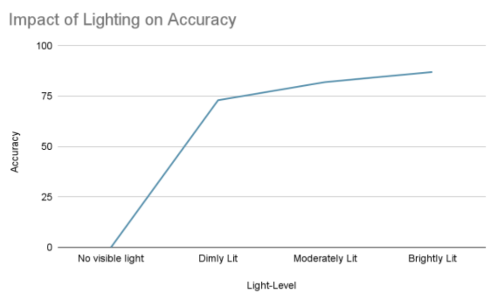

Figure 9. Impact of lighting on accuracy

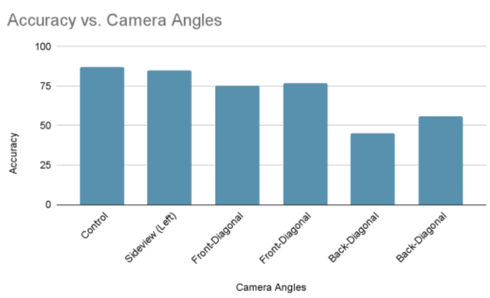

Figure 10. Accuracy vs. Camera angles

4.2. Experiment 2

Another potential blind spot in my program is the myriad of user experiences, perspectives, and impressions that could have possibly been affected through external elements. By collecting their experience, there could be further improvements made to enhance this product.

For this experiment, feedback from the users will be collected after providing the application and necessary equipment and allowing them to use it throughout the week. The survey itself will be designed to gather information both about the user's posture history and also the means through which they interacted the most with the system. We will be looking primarily at comfort derived from posture as well as the subject's ability to focus in terms of impact. There will also be a field where respondents can provide general suggestions to allow for a more nuanced picture of the experience.

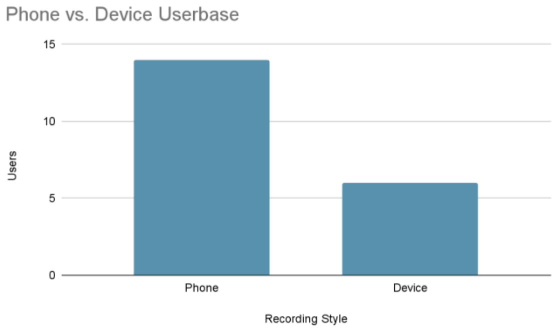

Figure 11. Phone vs. Device userbase

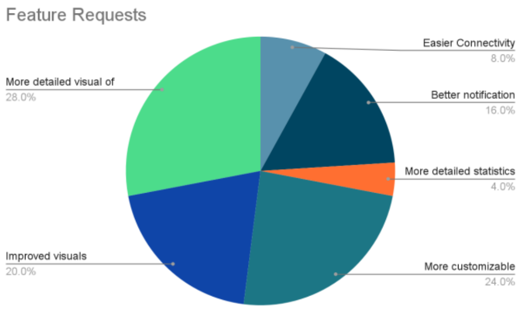

Figure 12. Feature requests

5. Related Work

A study conducted by Lauren Simpson, Monish M. Maharaj, and Ralph J Mobbs examined the role of wearables in spinal posture analysis. Wearables are technologies worn on the body and measure step count, heart rate, sleep quality, and other parameters. They are shown to have the potential to monitor posture to prevent further neuromuscular consequences using biofeedback. The accuracy of these devices was high with error ranges within 5 degrees of the main capturing moments [10]. However, their biggest challenge remains in the lack of validation as most reports only experiment on small samples of these devices over a short period of time. In addition, the practicality of these technologies may also be questioned. They require an optimum number of 3 sensors for an accurate reading of posture analysis data. With multiple sensors attached, this may pose a challenge for users to remove and reattach when needed, reducing the practicality of it. In contrast, this project only requires the user's phone camera or a Raspberry Pi camera module to use.

Qilong Wan et al conducted a study observing hip positioning and sitting posture via the use of a pressure sensor tied to a novel hip positioning algorithm. The system they developed can accurately locate hip angles and pressure values to classify four types of sitting postures. This is done using a Support Vector Machine with a polynomial kernel and can achieve an accuracy rating of about 89.6% [5]. This study also targets the need for better posture guidance to improve chair ergonomics. The one disadvantage here is that their solution requires a custom array of sensors that the user needs to sit down on, which is not as readily available as a smartphone or small portable device like what we are proposing.

Researcher Change et al took a more general approach to the problem by discussing the overall design of intelligent products for sitting posture correction. They point out how “Improper sitting posture can also indirectly cause myopia. A research report from the World Health Organization shows, that in 2020, the number of global myopia patients is about 2.5 billion” [14] The initial device prototype they show off is a Lego EV3 model that the user straps to various parts of their body to gather positional and posture data paired with notifications and reminders. This is similar to our approach of sending reports and reminders for people with bad posture, with the main difference being that we are pushing for a computer vision-based system that does not require the user to strap anything on them.

6. CONCLUSIONS

Multiple limitations are present in this project, starting with its hardware requirements. The camera module or phone must be mounted on a high enough platform, otherwise it will be difficult for it to view and analyze the user. Furthermore, sufficient lighting is also recommended for improved clarity and quality of the recording to support the accuracy of the AI.

The software limitations mainly fall under the issue of speed. This is because, between every frame of the live stream, there is some amount of time that must be spent identifying the landmarks and analyzing the data. This creates an issue for live feeds as spending too little time would result in inaccurate results while taking too long would result in laggier-looking video streams.

The implementation of a faster landmark detection model combined with a more sophisticated implementation of the analysis algorithm given the landmark data can help further improve the usability of the system without compromising on accuracy. A small periodic stream delay could also be employed to give the live feed enough time between processing frames to give a smoother playback when the user is briefly viewing them.

A general hardware improvement that can be made on the hardware side is through the use of an IR light combined with a camera capable of perceiving it. Infrared light is invisible to the naked eye but can help provide illumination for a camera even in the dark. Of course, this particular improvement applies more to the dedicated device, although a small add-on accessory for a phone could be an option to consider.

Our solution serves as a valuable tool for enhancing and promoting optimal posture habits. Through the ever-improving AI, this solution can contribute to long-term posture improvement and overall well-being with its simplicity and accessibility. Ultimately, incorporating this product will become increasingly relevant and contribute to a health-conscious community.

REFERENCES

[1] Yang, Lei, et al. "Prevalence of incorrect posture among children and adolescents: Finding from a large population-based study in China." Iscience 23.5 (2020).
[2] Ludwig, Oliver, et al. "Age-dependency of posture parameters in children and adolescents." Journal of Physical Therapy Science 28.5 (2016): 1607-1610.
[3] Robbins, M., I. P. Johnson, and C. Cunliffe. "Encouraging good posture in school children using computers." Clinical Chiropractic 12.1 (2009): 35-44.
[4] He, J., et al. "IOP conference series: materials science and engineering." Proceedings of the International Conference on Functional Materials and Chemical Engineering, Dubai, United Arab Emirates. 2017.

[5] Wan, Qilong, et al. "Hip positioning and sitting posture recognition based on human sitting pressure image." Sensors 21.2 (2021): 426.
[6] Wang, Hong, et al. "Quantifying the impacts of posture changes on office worker productivity: an exploratory study using effective computer interactions as a real-time indicator." BMC Public Health 23.1 (2023): 2198.
[7] Li, Miaoyu, et al. "Sitsen: Passive sitting posture sensing based on wireless devices." International Journal of Distributed Sensor Networks 17.7 (2021): 15501477211024846.
[8] Jung, Sang In, et al. "The effect of smartphone usage time on posture and respiratory function." Journal of physical therapy science 28.1 (2016): 186-189.
[9] Susilowati, Indri Hapsari, et al. "The prevalence of bad posture and musculoskeletal symptoms originating from the use of gadgets as an impact of the work from home program of the university community." Heliyon 8.10 (2022).
[10] Simpson, Lauren, Monish M. Maharaj, and Ralph J. Mobbs. "The role of wearables in spinal posture analysis: A systematic review." BMC musculoskeletal disorders 20 (2019): 1-14.
[11] Zemp, Roland, et al. "Occupational sitting behaviour and its relationship with back pain–A pilot study." Applied ergonomics 56 (2016): 84-91.
[12] Kamiya, Kazuhiro, et al. "Sitting posture analysis by pressure sensors." 2008 19th International Conference on Pattern Recognition. IEEE, 2008.
[13] Hazar, Zeynep, Gul OznurKarabicak, and Ugur Tiftikci. "Reliability of photographic posture analysis of adolescents." Journal of physical therapy science 27.10 (2015): 3123-3126.
[14] Wang, Yi. "Research and Application of Sitting Posture Correction Design Based on Artificial Intelligence." 2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM). IEEE, 2022.
[15] Kallings, Lena V., et al. "Workplace sitting is associated with self-reported general health and back/neck pain: a cross-sectional analysis in 44,978 employees." BMC Public Health 21.1 (2021): 875.