UTILIZING DEEP MACHINE LEARNING TO CREATE A CONTEXTALLY AND ENVIRONMENTALLY AWARE APPLICATION TO PREVENT SPINALTENDONITIS

Barry Li[1] and Yu Sun[2]

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

ABSTRACT

Recently, we have discovered when a person is using their computer, they often begin to lean forward toward the screen without noticing. Leaning forward can cause many problems in their body, especially to the back bone known as spinal tendonitis, and the problem can spread throughout the entire body [1][2]. I created an app to warn usersto sit up straight when they lean toward the screen too much, effectively protecting them from damaging their back bone. This app uses deep learning to calculate the body posture, and draw an imaginary triangle between the shoulders, hips, and knees [3]. The point at the hips is most vital in calculating the angle of the body. This app takes pictures in a given interval of time (by default 30 second), when the body leans forward, this angle decreases, and when the angle becomes lower than a given amount (by default 30 degrees), it will send a warning message to ask the user to fix their sitting posture [4].

KEYWORDS

Machine Learning, Application, Spinal Tendonitis.

1. INTRODUCTION

Individuals who use computers consistently are more likely to notice their tendons and muscles, primarily in the back and neck, deteriorate over time. These problems are just the beginning as many start with pain in one region of the body that then spreads to different parts of their bodies affecting an individual's overall productivity. Correcting sitting posture has recently been popularized. In order to prevent leaning, someone needs to keep track of how the computer user is sitting and fix his or her posture when leaning forward. However, no person can monitor the computer user for a long time, as they may have some other tasks to do. Therefore, it is important to develop an app able to detect the sitting posture and send a warning when bad sitting posture has been detected. The benefits of using this app are being able to correct bad sitting posture and having a better understanding of how often a person leans forward while using their computer. However, the consequences of using this app include misdetection when the app sends a warning message when the subject is not leaning forward. In addition, the app cannot always catch all cases of the person leaning forward. This can result in-app notification spams if the person does not fix the sitting posture which could in turn affect the user's working environment.

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 131-137, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122009

Some pose estimation techniques and systems have been proposed to detect the sitting posture; however, these proposals assume the user is sitting on a chair upright, which is rarely the case in practice. The estimation implementations are also limited in scale, with samples only given for sitting on the ground or bending knees upward. Other techniques, such as calculating the angle between the imaginary triangle, is used to determine if the person is leaning or not. To reduce complexity, the current algorithm used in these techniques cannot be too sophisticated and often results in errors in processing the picture and the angles of the sitting position. A second practical problem is that some users find it hard to understand how to use these methods, such as where to put the camera and the optimal application for maximum effectiveness of the detecting method.

In this paper, we follow the same line of research by finding how to detect a person's sitting posture [5]. Our goal is to figure out if a person is leaning too far forward or not. Our method is inspired by deep learning to detect the pose of a person. In order to do this effectively and efficiently, the program will locate points of the sitting person's body, mainly by machine learning to determine where the person usually sits and the location of their body parts. Second, it will locate the person's ears, hips, and knees and remember those points. Third, the app forms a triangle with the three points and calculates the angle of the point at the hips to get the angle measure result. Therefore, we believe that the calculation of the angle can help detect the sitting posture of a person and decide if he/she is leaning or not. This helps by figuring out how a person is sitting in their seat and if they are bending or not. If a person is found leaning forward, by notifying the angle is low enough, it will send a warning to tell the user to fix their sitting posture [6].

In two application scenarios, we demonstrate how the above combination of techniques increase the accuracy of finding out the sitting posture and the effectiveness of the detecting system. First, the usefulness of our approach will be proven by a comprehensive case study on the evolution of AI and deep learning to define where each part of the body of the user is located [7]. Second, we will analyze the evolution of motion detection in calculating the sitting posture of the person sitting in his/her chair. The coordinates of the points marked are accessible using the API provided and readable by code. An equation is used to find out what angle the person is leaning. The equation contains division ensuring a zero (0) is not allowed in the denominator. If the denominator produces a zero, the calculation for this image is automatically canceled and a warning message is returned to the end user. Whenever the AI fails to mark all of the points required for calculation, the equation will be automatically voided and will assume the user did not lean too much forward even if the person does. The AI still have some problems of detecting the person's body components and often result in not marking the points correctly due to environmental factors that can misunderstand the AI's detecting knowledge, such as too bright or dark, similar color the the person's clothing to the objects around, and objects blocking the camera's vision. We are still finding ways to overcome these obstacles and increase the accuracy of the camera. The camera we used does not have a high resolution and makes the AI somewhat harder to see things correctly.

This paper will be describing the project in more detail and will be organized in the following structure: The next section described the details on the challenges that we met during the experiment and designing the sample; Section 3 is focused on our solutions to overcome the challenges that we encountered and mentioned in Section 2; Section 4 presents how each challenges are overcome step by step, and followed by presenting the related work made by other people in Section 5. And, finally, Section 6 concludes this paper by briefly going over the contents on the paper again, and pointing out the future works that we be do to improve the project.

2. CHALLENGES

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

2.1. Building, installing, and setting up the Raspberry Pi

The first challenge I encountered was building, installing, and setting up the Raspberry Pi. The first step was to buy all the parts necessary to build the Raspberry Pi: A Raspberry Pi computer with an SD card or micro SD card, a monitor with a cable and HDMI adapter, a USB keyboard and mouse, and a reliable power supply. After reading a quick tutorial, the build went pretty smoothly which solved most of the hardware issues. Unfortunately, the software installation was much more challenging. I quickly found out that I was unfamiliar with the Linux Debian-based operating system [8]. I used YouTube tutorials to guide me through how to use the Linux operating system, and, after some basic command prompts and reboots, the single board computer was ready to use. In order to set it up, I need to set it up with the remote control. I installed VNC viewer, an app on Windows computers to access Raspberry Pi remote control using wifi.

2.2. How to use AI

Another challenge I faced was researching how to use AI to detect where each body part of the user is located in the image. After capturing an image of the subject's entire body, the app automatically displays the coordinates of the triangular points on the image. Lines are then virtually drawn between the coordinates of the points given. Only the points that are very necessary are connected, and these points are then calculated in order to determine the path of the lines. The detection gives the computer data to analyze how the person is sitting and if the person is leaning too much forward. The module sometimes did not recognize the position due to the limited angle of the camera. An additional roadblock was the fact that if the color of the clothes was similar to the background then detection was often more inaccurate. We adapted the app to take a video and picked out the best frames from video instead of only taking a picture. The AI image recognition then produced results with much better accuracy during the real-time process.

2.3. Setting the programming environment up

The final and last challenge was setting the programming environment up. The operating system of this Raspberry Pi is Rasbian [9]. It is called Rasbian because it is an alternative version of the Debian system, and, therefore, most of the reliability libraries cannot be easily installed with pip. Pip is a library that contains codes that are used for taking images. In order to solve this problem,I need to look for different wheel files to identify any problems or errors that may occur during programming. Wheel files are used to determine the function and execution of each pip library. If there is a slight mistake in the coding of the wheel files, then the pip library will also fail, so it is essential to ensure the wheel files are set up correctly before continuing with any additional programming. For each step taken, I used research to find out how calculation and code works, and used my own programming skills to perfect the code.

3. SOLUTION

After successfully conquering the three challenges above, I was able to create a system that efficiently and effectively determined the sitting position of the user. At first, when the camera is set up, it is connected to Raspberry Pi and ready to take pictures. I then run the program in the computer Linux terminal, but not the IDE terminal because TensorFlow does not work in IDE as the IDE terminal is not allowed to modify files and TensorFlow requires the plotting the dots and drawing the lines [10]. The TensorFlow function also returns the coordinates of the points marked relative to the image, which can be used for more purposes than just marking on the image. The dataset of the coordinates of the points marked are later put into the looping function that runs the calculation of the angle located at the hips. I found the equation online that results in the angle of one point, and assumed the hip to be that point and result in the angle. However, the dataset can sometimes not contain the coordinated required points either because it is out of image or blocked by another object in between the camera and the person. In response, I decided to implement the try function that will let the compiler verify if the code in this segment is able to run (contains all the points needed) or not. If any required points are found missing, the try function will determine it not to work and skip the function, I default this to a normal sitting posture if the equation will not run. If the angle is found to be less than 60 degrees, the alert message will appear, asking the user to realize their sitting posture and fix it back to the upright position.

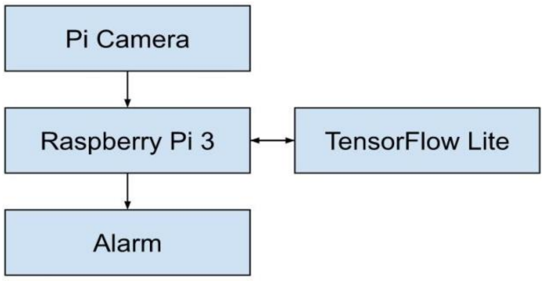

Figure 1. Overview of the solution

I used a Raspberry Pi Camera Module 2 and a code library so the images captured from the camera would automatically save onto the Raspberry Pi computer. In order to increase the accuracy of the position calculations I then instructed the camera to capture requires a camera installed to the Raspberry Pi to take videos instead of pictures then draw lines that connect the three major points plotted between the user's ears, hips, and knees. In addition to the hardware components such as the Raspberry Pi and Camera Module 2, I also used machine learning from TensorFlow to mark the body components that are used to define the angle of the person sitting. It can also mark other main components, such as eyes, nose, and feet, but those are not necessary for the sitting pose application. While the camera on my Raspberry Pi console is low and sometimes cannot see that well, it caused some problems while I'm testing my app. The program will send the message to the user's screen when leaning too much, prompting the user to fix their sitting posture. It currently only shows up in the computer's command console and does not always appear at the window in the top layer, it can be blocked by other windows. However, when used in modified applications, it is possible to show the message in a new window that pops out when bad sitting posture is detected. I developed this application using a python IDE canned Thonny Py, it is simple to use and, suitable for some advanced code development. Some obstacles in using Raspberry Pi are low performance, the demand of online searches, and unfamiliarity with the operating system. The low performances are natural to Raspberry Pi since it is a small computer that uses low-end hard-wares, this causes some limitation on the variety of the application that can be created, but it is suitable for the sitting pose detection. Online searches are done for the resources needed and how to build and use the code. They are essential for the creation of usable code and to modify the code to make it fit for the application. I'm also not familiar with Rasbian, or the operating system of Raspberry Pi, which I need to do research on how to use the operating system. Rasbian mostly uses the controls of Windows, but its terminal is based on Linux, which I'm not familiar with, for example, when I used ctrl+c and ctrl+v it didn't work and I must use right click and select from the menu to operate copy and paste. The console can perform actions that cannot be done inside the programming IDE and are often run by a shortcut on the main screen.

4. EXPERIMENT

4.1. Experiment 1

At first, I used a Pi Camera to take the picture, but it was blurry and tinted reddish, this was because of the specification and the lens of the camera. I first took a picture of myself and found out that the lines are drawn incorrectly and some dots are missing because of my camera. However, if the person is visible enough, the program works as it is intended and draws lines along my body. It shows TensorFlow can actually be used to detect the posture of the person sitting and know the angle that the person is leaning too forward or backward.

After the computer draws lines on the image using TensorFlow, it appears to give out the coordinates of the points on the image. The coordinates can be used to fill in an equation to calculate the angle of the person is leaning. The equation is being used to calculate the angle of the three dots formed as a result of the angle taken on the hips. I first tested out sitting straight up and it showed my angle of tilt is about 90 degrees, which is the perfect angle. While also some pictures, since not all three points are marked, did not fulfill the inputs the equation needed, and the program crashed. The program often said that I cannot divide by zero and the program will not be working consistently. The error was due to the missing number of the points taken since some parts of the body cannot be analyzed.

I later rewrote the code to prevent the error from affecting the code running and stopping the code. I added the try method to catch the error if the code is interrupted by division by zero. With the new code in place, even if not all three dots are marked, the program will not crash, but rather not show the message if the person is leaning too much forward. It can keep the program running even after a point is missing, and prevent it from crashing. It also takes work from providing an alternative procedure for the code to execute when it is not able to operate properly.

When I run the test again, the program works as it is intended, it warns me when the angle is too low, or lower than 60 degrees. The program keeps running without crashing when something wrong with the marking point happens. It can be configured to show an error message on the console while still running, and notice other developers that something is wrong, such as not all required points are marked, asking them to consider testing in a better environment that is more free from being blocked by objects such as the chair. When I fix my back position to upright, the message disappears and is no longer sent in the console. However, the display window is still not implemented so the message still appears in the console, which can sometimes be blocked by other windows open. I'm still finding resources to help me implement the displaying of the warning message in a separate window.

When I find ways to build and install Raspberry Pi, I first build my Raspberry Pi computer with my instructor to install the parts of the computer into its provided shell. To set Raspberry Pi up, I followed the instructions given by my instructor and on the display screen of my Raspberry Pi to install the operating system. I later take some time to explore and get to know how to use the newly installed Raspberry Pi and its operating system. Once I started designing the application, I realized that I will need an AI library installed to take pictures and mark the body parts. I researched on the internet for an AI usable by Raspberry Pi to detect the person's sitting posture, and I found TensorFlow as my option. TensorFlow allows me to detect how a person's posture is, like standing, sitting, or anything that involves moving. I finally researched for the equation to calculate the angle of the person sitting and made the code function as I tendered to.

5. RELATED WORK

Poor sitting posture has led to a variety of spinal disorders, per the research on A sitting posture recognition system based on 3 axis accelerometer [11]. It uses a tri-axial accelerometer to measure how a person is sitting by attaching the accelerometer on the back of the subject's neck to measure 5 types of sitting posture. It uses an accelerometer attached to the person instead of a camera taking pictures of the person which mine does. It has a better ability to detect the person as it is attached to the subject directly instead of taking a picture and evaluating how the person is sitting.

A method to predict 3D positions of body joints is from a research called Real-time human pose recognition in parts from single depth images [12]. It involves mapping the difficult pose estimation problem into a simple per-pixel classification problem. These methods also useimages to determine if a person is leaning too forward or not, and makes use of the camera's estimation of the person’s body parts. However, it also involves turning the image into a simpler per-pixel detection than just marking points on a natural image. The program makes better use ofthe image detection and body mapping functions by giving the computer an easier reading of the image and the body.

Personalized services that improve the living environment of a person involve sitting posture, the project Sitting posture analysis by pressure sensors shows how to use pressure sensors on the seat to determine how a person is sitting [13]. The classification rate of an unknown person is about 93.9%, compared to about 98.9% for a known person. The project involves using a pressure sensor on the user’s seat that receives information by the weight of the person sitting on it, the weight changes when the person is sitting in different postures and can detect 9 different postures. The body weight measurement tends to be less accurate than using camera detection because body weight pressure on the sensor may not determine the sitting posture in an understandable way.

6. CONCLUSIONS

I have created a sitting posture detection app that involves taking pictures and using Tensorflow to determine how a person is sitting using markings on the body skeleton and the angle of the hips with the intersecting lines of the head and lower spines and the knees to the hips. The main purpose of my program is to solve the back spine getting injured while using the computer. I used a Pi Camera to take pictures and TensorFlow Lite to analyze the sitting posture of the person [14]. If the person is leaning too much forward, the angle located at the hips of the person is lower and sends a warning message when this angle reaches below 60 degrees to tell them to fix their sitting posture. I applied this method by an equation that calculates the angle ratio between the three points and returns the result, which is the point that is marking the hips. TensorFlow is used to draw out the person’s skeleton in the image and returns the coordinates of the dots marking in the image. The equation is researched on the internet and written separately from the libraries I downloaded, and involves using trigonometric functions. I also implemented the error catch when the equation attempts to divide by zero. The program sometimes does not work when at least one point that is required for the calculation is missing, often by something blocking the body from the vision of the camera. It can be effective at most times in detecting bad sitting postures but sometimes not doing so that well.

The limitations of my program includes the chance of a picture of the full person captured by the camera without anything blocking in the way, the detection of the lower spines are also critical toward a good sitting posture but cannot be implemented and well detected because of the limitation of the points drawn by TensorFlow, and the ability to launch by conventional ways, such as a shortcut on desktop or built into an application. The person can sometimes be blocked by other objects that reduce the proficiency of the camera detecting the person's sitting posture. And the built in app is not created because I lack the requirement of the knowledge of opening a new window with Python [15].

In the future, I will implement a better way of taking the picture taken and adding a pop-up window to warn the user to make the app easier to use. I should also recommend the distance between the camera and the person when the camera is detecting if a person is leaning or not. It makes the app much easier to use if these features are implemented and more functional to solve the health problem caused by leaning too forward toward the screen.

REFERENCES

[1] Nachemson, Alf. "The effect of forward leaning on lumbar intradiscal pressure." Acta Orthopaedica Scandinavica 35.1-4 (1965): 314-328.

[2] Artenian, D. J., et al. "Acute neck pain due to tendonitis of the longus colli: CT and MRI findings." Neuroradiology 31.2 (1989): 166-169.

[3] Deng, Li, and Dong Yu. "Deep learning: methods and applications." Foundations and trends® in signal processing 7.3–4 (2014): 197-387.

[4] Andersson, BJ Gunnar, et al. "The sitting posture: an electromyographic and discometric study." Orthopedic Clinics of North America 6.1 (1975): 105-120.

[5] Claus, Andrew P., et al. "Is 'ideal'sitting posture real?: Measurement of spinal curves in four sitting postures." Manual therapy 14.4 (2009): 404-408.

[6] Harbach, Marian, et al. "Sorry, I don' t get it: An analysis of warning message texts." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2013.

[7] Serag, Ahmed, et al. "Translational AI and deep learning in diagnostic pathology." Frontiers in medicine 6 (2019): 185.

[8] Emm anouil, Emmanouil, and Angelo Sifaleras. "Implementation of an Open Source Optimization Software Package for Debian-Based Operating Systems."

[9] Kurniawan, Agus. "Programming on Raspbian OS." Raspbian OS Programming with the Raspberry Pi. Apress, Berkeley, CA, 2019. 79-96.

[10] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.

[11] S. Ma, W. -H. Cho, C. -H. Quan and S. Lee, "A sitting posture recognition system based on 3 axis accelerometer," 2016 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2016, pp. 1-3, doi: 10.1109/CIBCB.2016.7758131.

[12] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," CVPR 2011, 2011, pp. 1297-1304, doi: 10.1109/CVPR.2011.5995316.

[13] Kazuhiro Kamiya, Mineichi Kudo, Hidetoshi Nonaka and Jun Toyama, "Sitting posture analysis by pressure sensors," 2008 19th International Conference on Pattern Recognition, 2008, pp. 1-4, doi: 10.1109/ICPR.2008.4761863.

[14] Symon, Aslam Forhad, et al. "Design and development of a smart baby monitoring system based on Raspberry Pi and Pi camera." 2017 4th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2017.

[15] Borcherds, P. H. "Python: a language for computational physics." Computer Physics Communications 177.1-2 (2007): 199-201.