HIGH SCHOOL EDITION
Journal of Student Research

Volume 12 Issue 4 (2023)

Deep-Learning Based Automatic Ergonomic Assessment Using Webcam Data

Owen Lu[1], Dr. Clark Hochgraf[2#]

[1] Monta Vista High School, Cupertino, CA
[2] Rochester Institute of Technology, Rochester, NY
[#]Advisor

ABSTRACT

Primarily due to increasing computer use, people are spending more and more time sitting in front of a desk every day. However, prolonged sitting has been associated with tiredness, hypertension, and pain in areas like the lower back or shoulders. These symptoms arise for a variety of reasons, but musculoskeletal disorders in particular are largely associated with poor postures. The adverse results caused by poor postures can be controlled with proper training and monitoring. This study attempts to provide automatic ergonomic assessment using only webcam data. Since laptops, desktops, and phones are now widely available and equipped with built-in cameras, this solution is accessible and convenient for most people. More importantly, automatic posture assessment may help to prevent conditions associated with poor posture by giving reminders whenever improper posture occurs. To create our model, we make use of Mediapipe, which provides a solution to identifying keypoint locations from an image. By training our MLP classifier on this key-point data, we achieved a 96.96% test F1 score, indicating that our system serves as a convenient way to assess posture while maintaining high performance. To illustrate our results, we perform a final video classification by overlaying the model's pre-dictions on each frame.

Introduction

As people spend more time sitting for extended periods in front of their computers, desk ergonomics is becoming an increasingly important area to study. A study of working Australian adults found that they spend an average of 9.4 hours on weekdays sitting, over half of which occurs at work (Miller et al., 2004). Sitting by itself does not necessarily lead to musculoskeletal disorders (Lis et al., 2007), but awkward or bad posture results in a range of musculoskeletal disorders, leading to pain in areas like the lower back, shoulders, and neck. For example, sitting with a slouch could increase lower back discomfort (Jung et al., 2020), and having a prolonged forward head posture may lead to neck pain (Falla et al.,2007).

Unlike some other illnesses, musculoskeletal disorders resulting from improper posture largely stem from correctable habits, thus motivating the creation of a system to detect and warn an individual about these habits from early on. For instance, Mahmud et al. found that after receiving training for 12 months, workers had significant improvements in posture and had reduced pain in the neck and back (Mahmud et al., 2015).

With the rapidly advancing field of AI, supervised machine learning models can learn to detect such postures by training on data to learn on patterns that distinguish between postures that are more or less likely to contribute to discomfort. These models can effectively monitor and provide automated, real-time feedback on someone's posture to help reduce the risk of future health issues.

Researchers have tackled the task of sitting posture detection in two main ways. The first approach assesses posture by utilizing sensor data; once the sensor data is gathered and saved, a model is then trained on the data to detect posture. Because each study used different arrangements of sensors, the number of postures that their setup could detect also varied. For example, Arshad et al. (2022) used force-sensitive resistor (FSR) sensors along with ultrasonic sensors to detect four postures, namely leaning right, left, forward, and back. Bourahmoune et al. (2022) detected 15 postures using an arrangement of 9 pressure sensors, and Roh et al. (2018) detected six postures with four load cells positioned on the corners of a seat. Piñero-Fuentes et al. (2021) took a different approach and utilized video data taken from a camera. They first estimated the certain keypoint locations using a convolutional neural network, from which they calculated the angles between select pairs of joints to detect poor posture. For example, to check for shoulder alignment, they computed the angle between the left and right shoulder joints. The angles were then categorized into four groups, which rank the severity of the bad posture.

Our work addresses the following question: can subtle changes be accurately detected from only webcam data? Previous works either rely heavily on sensors, which require an intricate and costly setup, or rely on image and video, but need huge amount of training data and can only categorize few groups. Our work takes a different approach that utilizes image or video data only and be able to categorize eight key postures. The main contributions in this paper are: 1) combine Mediapipe into our model to get keypoint landmarks. 2) Propose six pair wise distances between landmarks and feed them into our dense network model for training. 3) Create a framework that can label webcam videos. Such a system is portable yet accurate to assess people's work posture. It serves as a viable option to anyone that has access to a camera, whether through a computer, a cell phone, or an external webcam. Furthermore, the automated posture assessment provides a convenient solution to help prevent musculoskeletal disorders associated with extended computer use.

Materials and Methods

Overview

Instead of feeding the raw images directly into a deep learning model, we first make use of the functionality present in the Mediapipe pose classifier, which offers a machine learning (ML) pipeline that outputs the 3D locations of 33 landmarks on an individual's body from an image. After extracting the locations of these 33 landmarks through Mediapipe, we pre-process and train a fully connected neural network on this data. Our design consists of the elements shown as Figure 1.

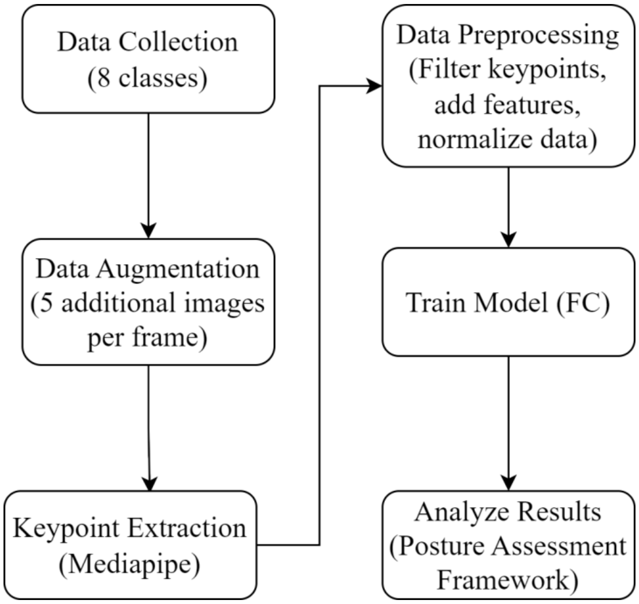
Data Collection (8 classes)
Data Augmentation (5 additional images per frame)
Keypoint Extraction (Mediapipe)
Data Preprocessing (Filter keypoints, add features, normalize data)
Train Model (FC)
Analyze Results (Posture Assessment Framework)

Figure 1. Project Flow. The six major components of the project are data collection, data augmentation, keypoint extraction, data preprocessing, model creation, and analysis of results.

Data Collection

After gauging the abilities of Mediapipe, we settled with eight classes as shown in Figure 2 for assessment: Upright, Shoulder Left, Shoulder Right, Leaning Forward, Leaning Backwards, Trunk Left, Trunk Right, and Head Forward.

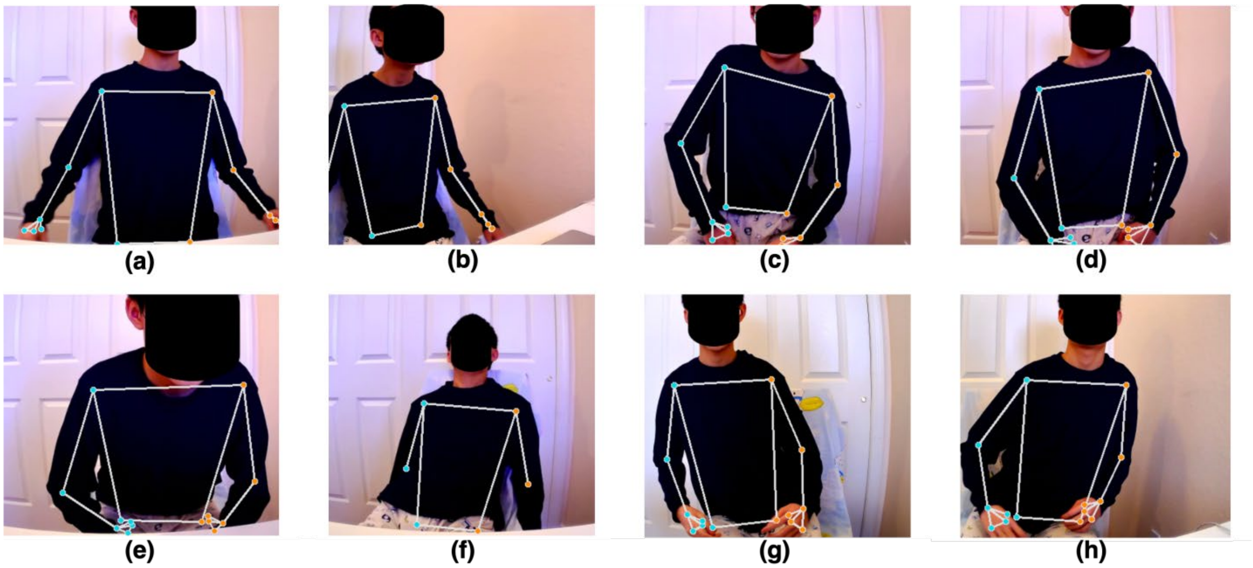

Figure 2. Posture classification. The eight postures our system can classify with keypoints drawn from Mediapipe: (a) Upright (b) Head Forward (c) Shoulder Left (d) Shoulder Right (e) Leaning Forward (f) Leaning Backwards (g) Tilting Left (h) Tilting Right.

Our setup consists of a webcam positioned right behind a computer located around 2 feet away. The webcam is then raised so that everywhere from the head to the hips is visible. To collect data for each class, a video was taken while that specific posture was held to ensure consistency and eliminate the need to manually label individual images; furthermore, to create more variety in the dataset, we rotated the camera horizontally with amplitude of around ±20 degrees, giving the model multiple angles of each posture to train on. Having variation in the dataset helps capture more generalized poses so that the model doesn't over fit and can predict from multiple camera positions. Postures are still recognizable under rotation because they depend on the relative, not absolute, positioning of the keypoints.

Realistically, most postures will have a combination of two or more of the above classes: for example, a person may lean forward and tilt their shoulders at the same time. Thus, multi-label classification also serves as an important goal for our model to achieve. Once the neural network learns the distinguishing features of the individual classes, it can predict on a more complicated image containing multiple postures by separately determining the image's similarity to one particular class and then combining the predictions for the multi-label classification.

Data augmentation

The data collection process creates one video per posture. For each posture, the video is then converted into images by extracting the individual frames at 10 frames per second, and the resolution is lowered from a 1080 x 720 to a 540 x 360 image. Lowering the resolution of the image serves as an efficient way to save storage without interfering too much with Mediapipe's estimation.

Even with the videos, we only had a few hundred frames for training after setting aside a portion for the test set. Thus, we perform two types of augmentation on the training set: translation (up to 20% of the height and width) and rotation (with a rotation range of ±10 degrees). The augmentation generates five additional images per frame, increasing both the amount and variety of data that the model to train on.

Before augmentation, the training set consisted of 4626 images; after augmentation, our dataset expanded to 27756 images. The image breakdown per class after augmentation is shown in Table 1.

Table 1. The dataset collected for our study.

Posture	# of Images for Training	# of Images for Testing
Head Forward	3330	51
Lean Forward	3588	51
Shoulder Left	3702	51
Shoulder Right	3186	51
Tilt Left	3522	51
Tilt Right	3264	51
Upright	3690	51

Data Preprocessing

Our data was preprocessed in four steps: 1) excluding images without valid keypoints from the dataset: this step is automatically performed by Mediapipe when outputting the pose landmarks. Mediapipe has two ways of evaluating the success of detection: first, it has a minimum detection confidence which tells us if the person was even detected at all. Second, if Mediapipe detects the person, it then continues to track the landmarks and uses another metric, the minimum tracking confidence, to determine if it detected the landmarks successfully. In our model, we used the default threshold of 0.5 for both metrics, which resulted in a removal of 651 images from the augmented dataset, or a 2.35% decrease. 2) Filtering out keypoints like foot that have little relevance to the postures we are classifying: for classification of shoulder posture, lower-body keypoints have little relevance; in fact, including such values actually harm the model and distort our results because the model may unintentionally train on these irrelevant values instead of the intended keypoints. Furthermore, since our images only captured the upper body, Mediapipe must infer everything below; with zero information about how the person is oriented, such inferences will probably be far off from the actual value, thus confusing the model even more. Hence, we decided to filter out irrelevant keypoints to ensure that our model trains on the right features. 3) Adding six additional features through calculating pair wise distances between landmarks: the (x, y, z) coordinates of the landmarks can be quite abstract. Including pair wise distances may help because by taking differences in the coordinates, the model can directly compare the distances to obtain a sense of the relative positioning of the keypoints: for example, if someone raises their shoulder to the left, the distance between the left hip joint and the left shoulder joint would be greater than the distance between the right hip joint and the right shoulder joint. Regardless of what the exact y-coordinate of these joints are, the distance only takes the difference in these coordinates into consideration, which provides a more useful quantity to train on. 4) Data normalization: to help the model converge, we standardized the values for each feature in the input data. For each column, the mean μ and the standard deviation σ are computed, and the standardized values are given by $(x - \mu)/\sigma$, where x is the original value.

The Model

After pre-processing our data, we train a fully connected neural network (FC) as in Figure 3 with the goal of accurate posture classification. Because we augmented our data, we split the data before augmentation to prevent repeated (or highly similar) images between our training and test sets.

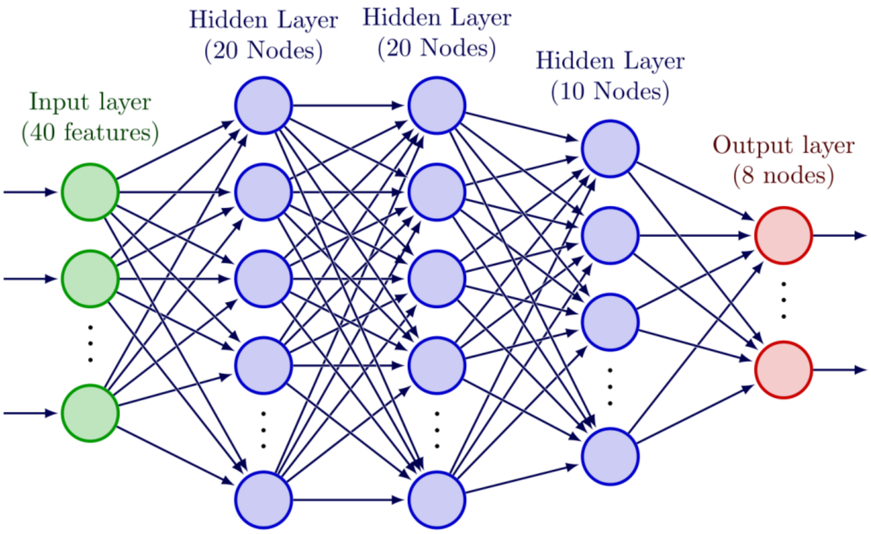

Figure 3. FC network structure. The model consists of an input layer, three hidden layers of sizes 20, 20, and 10, as well as an output layer that gives the prediction probabilities for each of the eight postures. The ReLU activation function was used between all layers except for the last layer, using a softmax activation.

The FC used for this study consists of three hidden layers with sizes 20, 20, and 10. We set our initial learning rate to 0.001 and the max number of iterations to 250, which were enough to ensure that the model converged smoothly.

Training the Model

To train our model, we used the sparse categorical cross-entropy loss, which is defined as

$$\mathcal{L} = -log\hat{y} \qquad (1)$$

where $\mathcal{L}$ denotes the loss for a single training example and $\hat{y}$ denotes the output probability for the expected class y. A ReLU activation function was used between each layer of the FC except for the last layer, which used a softmax activation function with 8 classes. The model was optimized using the Adam optimizer.

To evaluate the performance of the FC, the "macro" multi-class F1 score was used. For binary classification, if TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives, then precision P and recall R are defined as

$$P = TP/(TP + FP), \qquad (2)$$

$$R = TP/(TP + FN), \qquad (3)$$

and the F1 score is defined as

$$F1 = 2PR/(P + R). \quad (4)$$

The “macro” F1 score is then defined as the average of the per-class F1 scores.

Results

Our model was able to achieve a training F1 score of 99.70% and validation F1 score of 99.63% after 250 iterations, so no obvious over fitting was observed. On the test set, which consisted of 51 images per posture that were completely separate from the training set, the model obtained an F1 score of 96.96%. Looking at the confusion matrix for the test set in Figure 4, most of the incorrect classifications were confusing the head forward class with the upright class. The learning curve reveals that the MLP converged smoothly within 250 iterations, achieving a final training loss of 0.0522 and a final validation loss of 0.0500.

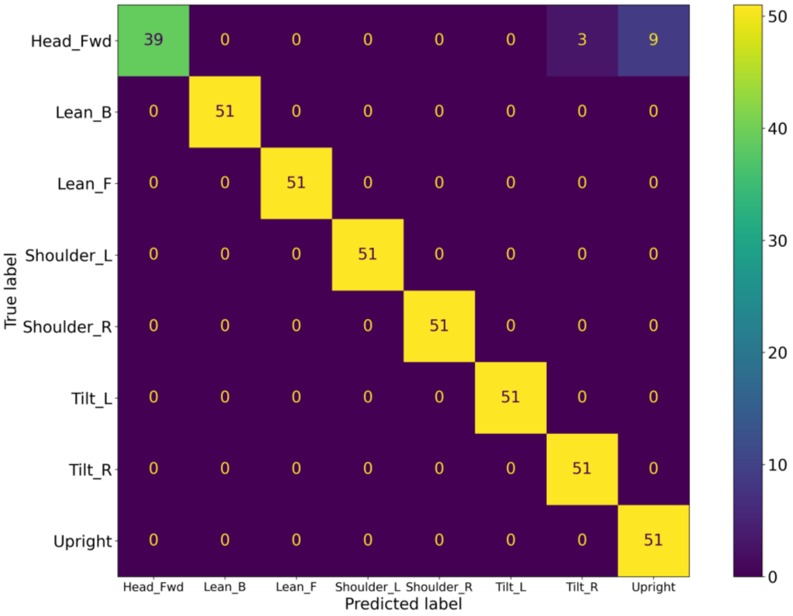

Figure 4. FC training results. After the FC converged on the training set, it was used to predict on the test set. The model's performance on the test set is illustrated in this confusion matrix.

With the trained MLP classifier, we were able to set up a framework that takes in a video and labels it with the annotated pose classification. Such a framework can help us see how the model performs in the long-term over multiple different postures. The framework works in the following manner: 1) split the video into individual frames; 2) for each frame, use Mediapipe to export the landmarks; 3) preprocess and predict the pose from these landmarks using the trained model; and finally 4) label each image and re-assemble a video from the images. To ensure the video was not used in training, we took a separate video and labeled it. A few screen captures from the final video are shown in Figure 5.

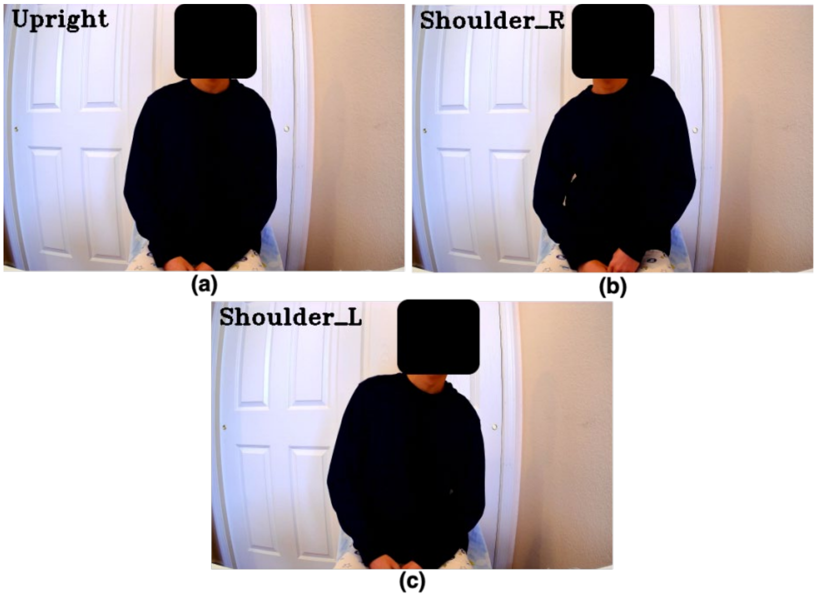

Figure 5. Webcam video with posture labels. A video containing multiple postures was taken as an input, and the model's prediction was overlaid on top of each frame before combining the frames back into a video. Screen-shots taken from the final annotated video when the model predicted (a) Upright (b) Shoulder Right (c) Shoulder Left.

Discussion

Other works have utilized sensors for pressure sensing as a base for their machine learning model, a setup that is costly and may require time to set up. Our model achieves a similar accuracy with just a webcam and consistently detects a wide range of postures, some of which cannot be detected by a chair sensor. For instance, our method of extracting the 3D coordinates can detect keypoints on the head so long as the head is in frame, while a chair sensor may not necessarily even reach the head or neck area.

Using pre-trained Mediapipe landmark labeling network greatly reduced the training cost to identify eight postures from raw images. However, it is crucial to properly and effectively use this publically available pre-trained model. Adding six pair wise distances between key landmarks is critical in our application. During model evaluation, we identified multiple inconsistencies with Mediapipe itself, which might have influenced our prediction. For instance, Mediapipe's inference of the z-coordinate of a landmark is often inaccurate, which may harm our predictions in classes like leaning forward or upright or backwards that are dependent on the z-coordinate. Interestingly, our model still achieved a high accuracy in these classes, indicating that the model successfully identified other factors correlated to the degree of leaning.

This work successfully created a model that accurately classifies sitting postures through webcam data. After generating a dataset containing Mediapipe keypoint locations extracted from the original image, we trained a MLP classifier to classify eight key postures.

Because our methodology uses landmark locations as opposed to the raw image, the detection of posture is less dependent on factors like lighting and background. However, while reviewing landmark labeling results, we found that occasionally, the camera rotated too much, removing several keypoints from the frame. Furthermore, the augmentation process, most notably translation, also removed a few keypoints, which Mediapipe had to infer instead. Despite the loss of important keypoints, the model still achieved high performance, indicating that the model trained on different features than what was intended. This shows the robustness of our deep learning method.

Our current model sometimes mixed up classes with supposedly little in common. For instance, on a test set created with the same background but from a slightly different location, the model predicted Tilt_R when the true label was Upright. After reviewing Mediapipe's annotated pose, the issue did not seem to come from a failure to accurately detect the keypoints. Thus, such a misclassification indicates that instead of learning to observe the position of the shoulder joints relative to the hip joints, the model instead trains on another lurking feature that is present in both the Tilt_R and Upright classes. As a hypothetical, if the individual was consistently looking up during these two classes, the model may have identified that similarity as opposed to tracking the location of the shoulder joints.

Despite the model's high F1-score on the training and test sets, it had some limitations. Most notably, the model's performance depends on the camera angle and position. Although we assumed people's sitting location variation related to the webcam is limited and tried to cover the variation while collecting our training data, more varied locations would have helped. Additionally, future work should look into ways to check which features are most heavily weighted in the result and check if the model's decisions make sense.

Finally, the framework that we currently have in place right now to predict posture is rather cumbersome: we have to upload a video, generate the landmarks, and pre-process the data before we can run it through the model. Thus, real-time classification app serves as the next benchmark in terms of automating the classification process.

References

Miller, R., Brown, W. Steps and sitting in a working population. Int. J. Behav. Med. 11, 219–224 (2004). https://doi.org/10.1207/s15327558ijbm1104_5

Lis, A. M., Black, K. M., Korn, H., & Nordin, M. (2007). Association between sitting and occupational LBP. European spine journal : official publication of the European Spine Society, the European Spinal Deformity Society, and the European Section of the Cervical Spine Research Society, 16(2), 283–298. https://doi.org/10.1007/s00586-006-0143-7

Jung, K. S., Jung, J. H., In, T. S., & Cho, H. Y. (2020). Effects of Prolonged Sitting with Slumped Posture on Trunk Muscular Fatigue in Adolescents with and without Chronic Lower Back Pain. Medicina (Kaunas, Lithuania), 57(1), 3. https://doi.org/10.3390/medicina57010003

Falla, D., Jull, G., Russell, T., Vicenzino, B., & Hodges, P. (2007). Effect of neck exercise on sitting posture in patients with chronic neck pain. Physical Therapy, 87(4), 408–417. https://doi.org/10.2522/ptj.20060009

Mahmud, N., Kenny, D. T., Md Zein, R., & Hassan, S. N. (2015). The effects of office ergonomic training on musculoskeletal complaints, sickness absence, and psychological well-being: a cluster randomized control trial. Asia-Pacific journal of public health, 27(2), NP1652–NP1668. https://doi.org/10.1177/1010539511419199

Arshad, J., Asim, H. M., Ashraf, M. A., Jaffery, M. H., Zaidi, K. S., & Amentie, M. D. (2022). An Intelligent Cost-Efficient System to Prevent the Improper Posture Hazards in Offices Using Machine Learning Algorithms. Computational intelligence and neuroscience, 2022, 7957148. https://doi.org/10.1155/2022/7957148

Bourahmoune, K., Ishac, K., & Amagasa, T. (2022). Intelligent Posture Training: Machine-Learning-Powered Human Sitting Posture Recognition Based on a Pressure-Sensing IoT Cushion. Sensors (Basel, Switzerland), 22(14), 5337. https://doi.org/10.3390/s22145337

Roh, J., Park, H. J., Lee, K. J., Hyeong, J., Kim, S., & Lee, B. (2018). Sitting Posture Monitoring System Based on a Low-Cost Load Cell Using Machine Learning. Sensors (Basel, Switzerland), 18(1), 208. https://doi.org/10.3390/s18010208

Piñero-Fuentes, E., Canas-Moreno, S., Rios-Navarro, A., Domínguez-Morales, M., Sevillano, J. L., & Linares-Barranco, A. (2021). A deep-learning based posture detection system for preventing telework-related musculoskeletal disorders. Sensors, 21(15), 5236. https://doi.org/10.3390/s21155236

MediaPipe Pose. https://google.github.io/mediapipe/solutions/pose.html