AutoPose: Pose Estimation for prevention of musculoskeletal problems using LSTM

*Note: This is a journal paper in progress

Francesco Bassino
Computer Science
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
u201816649@upc.edu.pe

Cesar Mosqueira
Computer Science
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
U201910750@upc.edu.pe

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Office work is one of the most common jobs in our society, where we can spend up to more than 8 hours sitting in front of a computer, laptop, tablet or notebook. In 2021 the average number of Europeans using a computer for work was 58% with variations between 37% and 85% depending on the country surveyed. In just 10 years the average growth per year of computer use in the workspace increased by 14%. If we estimated that this number is only going to increase (as it has been doing for a decade), it can be concluded that the number of people who end up suffering bodily pain from sitting is also going to increase [4].

Sitting for prolonged working hours can cause various fatigues both mentally and physically. In a study of 447 Iranian office workers, it was found that 48.8% of the participants did not feel comfortable at their workstation and 73.6% felt exhausted during the day. Also, 53.5% suffer from neck pain, 53.2% from low back pain and 51.6% from shoulder pain [3]. Apart from muscle discomfort, it also causes increased pressure, friction and shear on the chair, and the rubbing of the chair against the skin for a prolonged time causes the skin to peel off, leading to severe pressure ulcers, leading to death [1].

To attack this problem we have to divide it in sub-problems, the difficulty encountered lays on the idea of an exponential number of ways to solve this problem based on the way that we're gonna solve each one of the sub-problems. We have to decide what tools you may or may not use. The first sub-problem we need to solve is how to estimate a pose, which can be fairly simple using tools like Open Pose[2], PoseNet[9] or Deepcut[17], which are APIs that are able to use pose estimation algorithms mainly using pre-trained models of CNN variations like Mask R-CNN[7], or straight up models like YoloV7[21] or others pre-trained CNN architectures. The complications come in one major factor: How can we determine when a person is in a bad posture or at risk of generating muscleskeletal problems in the future? For that we need a Machine Learning model capable enough to detected what posture is and why is it harming for the worker.

Similar solutions have been reached in different papers. These vary in the way they solve the problem, for example in [4] they used weight and motion sensors in a chair. However, this does not fit our solution, since implementing a chair requires more time and money. We are looking for a solution that is quick, inexpensive and easy to use. Other solutions point to camera systems, like the one we propose, but with a different approach. In the article [10] the authors implement a camera system with a MobileNet V2[18] neural network. However, all of the approaches proposed are not accurate classifying bad posture, being limited to basic movements and not accurately displaying the problem.

The key components of our approach are the following. We implement a LSTM model, which gives us the best accuracy and runtime from all the other models tested. We use the NAO Robot for the camera, which also gives us the ability to give the user feedback in case the worker is in a bad posture. Some complications may come when putting all the systems together and creating a application. Furthermore, the implementation and creation of a dataset that allows us to train and test our application and model.

Our main contributions are as follows:

- We have proposed a LSTM model for bad posture detection.
- We have implemented a dataset consisting of images of good and bad posture at different angles.

- We have implemented a system that allows to estimate pose and prevent muscleskeletal disorders (MSD) using the NAO robot.

This paper is divided into the following sections: In Section II, we review related work on preventing MSD and pose estimation. We then discuss relevant concepts and theories related to the background of our research and describe in more detail our main contribution in Section III. Furthermore, we will explain the procedures performed and the experiments conducted in this work in Section IV. At the end, we will show the main conclusions of the project and indicate some recommendations for future work in Section V.

II. Related Works

The seating pose correction has been a concern since the office work became more and more present in the modern life. That is because this kind of sedentary work leads to back pain and other diseases that encompasses the MSD concept. The posture correction goes back to 1,850 with the invention of the corset. However, our research focuses on the pose correction to prevent aforementioned MSD's. We were able to find some other similar solution. We found some others solution to relevant problems like classifying pose, detecting wrong seating postures, and pose estimation using cameras, waves (WiFi) or sensors.

In [14], the authors propose the development of a smart chair that can classify different seating postures. The chair uses 8 sensors placed in strategic places on the chair, each one can determine the pressure applied to them and then send them via WiFi to a desktop application. They were able to classify 8 different postures, to do that they did an experiment on over 40 subjects inviting them to seat on those positions. They trained a deep learning model that is able to classify that data with a 91.68% accuracy. Our idea is different than this one in a few ways since, we are using a pose estimation system based on the input of a camera instead of sensors and instead of classifying the poses with a training data, we rely on an LSTM which will be trained based on assessments[5] that have already been studied by the corresponding scientists of the area. This means that our solution will have a more robust way to proof if the output is right or wrong than a classifier.

In [11] they explain the importance of the preventing the Work-related muscleskeletal disorders. Mostly related to the kind of labor that leaks on ergonomic assistance like tasks found in hotels, factories, construction, assembling, and big etc. They use cameras for this solution and since they try to get a real time response the model they use for predicting this is a long-short term memory neural network (LSTMNN). The LSTMNN is a model that allows inputs of sequences of data like videos. In order to determine which pose is risky or not they use Ovako Working Posture Analysis System[5] (OWAS). Which are a set of assessments that can give a score of how good or bad the posture is and that's the output they try to get. Our approach is heavily inspired by this paper since it points out the importance of preventing MSD in a work related environment and it also explains various methods that we could use to solve this problem. Our solution also uses a single camera to detect the joint positions and then a trained LSTMNN to define the risk factor on each posture and compare them with the OWAS metrics.

In [8], the authors developed a model that receives input from a system that can determine the position of a person through wireless classifier. The model would be a Support vector machine (SVM) and they intent to classify 7 daily life activities. They manage to get a 95.4% accuracy. They use 4 layers and 6 classifiers, the first classifier determines if the subject is in movement or staying. The second classifier determines if the subject is running or walking. The third classifier determines if the movement in question is a body or joint caused. The forth classifier determines if the subject is seating or standing. The other two are used to detect if the movement is caused by a feet or arm and the last one classifies the frequency or repetitions of the movement. This proposal intends to prove that the human body can reflect certain frequencies of radio waves by detecting this daily life activities. This differs from our approach solution-wise, since we intend to prevent MSD by detecting seating postures, and theirs is developing a human activity detector that is robust to environmental changes (thus the WiFi approach). They use the machine learning model SVM and we're using a deep learning approach like is the LSTM. This means that the model they trained will be able to detect human activities with high accuracy since the WiFi model they built is able to detect the pose very accurately and from there the SVM will have an easier job. On the other side our solution using a camera will probably struggle estimating the pose but the LSTM should have an easier job since it'll use a sequence of 3 or 5 frames to process the final output.

In [23], the authors developed a model monitor sleep posture in patients. That is important because it can track the progression of Parkinson's patients and epilepsy patients that often sleep in fatal postures. To achieve this they used radio frequencies. Apparently the human body can act as a reflector in low GHz frequencies. They made an experiment on 26 different homes using 26 different subjects and more than 200 nights. They achieved a precision of 83.7% with only 16 minutes of data. The problem they're trying to solve can relate to ours. With the difference that we're trying to prevent problems that a bad posture can lead to when we're working in a desk and they're aiming to prevent problems that bad posture can cause when sleeping. The approach they took to solve the problem differs with ours in a similar way than [14], since they'll have an easier time estimating the pose. However since in this case they're trying to detect sleeping fatal postures, we believe that in this case it would be harder to validate the output of their model even with a larger dataset although, 83.7% accuracy is really good.

In [10], the authors developed a system that is able to predict a few different types of seating postures using just the webcam that records a front view of the user. They made a variation on MobileNetV2 allowing it to use video as an input by including recurrent layers on it making the network able to use sequences of frames to determine the type of posture that is being recorded using an unsupervised machine learning approach. Ours could be considered an improvement to this solution since we're gonna use the camera in a more strategic place than the web cam. That will give a better result. However, it might be harder or more uncomfortable to use in case it becomes a product. The paper present various comparisons and between different models, the main ones he compares are a Convolutional Neural Network (CNN), a LSTM, ResNetV2[6] a MobileNetV2 model, presenting confusion matrices and different accuracy values for different situations like the frame-rate of the camera in case they're using a recurrent neural network and different resolutions of the images in use. We will replicate this result showing process with a similar benchmark.

III. AutoPosture: Pose Estimation for Prevention of Musculoskeletal problems using LSTM

A. Preliminary Concepts

In this sections we will introduce in a more elaborated way the concepts we are going to use in the rest of this work. Mostly including deep learning, computer vision and ergonomics.

Definition 1: (Human Activity Recognition): HAR is a subfield of computer vision and machine learning that focuses on identifying and analyzing human activities using different methods for data collection such as cameras, sensors, radio waves, accelerometers, etc. The goal of HAR is to automatically recognize and classify human actions or behaviors in real-time, with applications ranging from surveillance and security to healthcare and sports performance analysis. This is a very broad definition and the we consider that the problem we're trying to solve is consider a sub-field related to HAR since we will use the same concept of HAR to monitor ergonomics in order to detect issues in posture.

Definition 2: (Joint angle estimation): In order to be able to detect MSD must have information about the current state of the human body. Most assessments to detect MSD require the angles between various joints in the body, such as the knee, elbow, and shoulder. With those we may be able to identify some abnormal patterns or postures that could lead to MSDs.

Definition 3: (Ergonomics): Ergonomics is the study of how to design workspaces, tools, and equipment to minimize the risk of injury or strain. By combining computer vision data with ergonomic principles, it may be possible to identify specific changes to workspaces or equipment that could reduce the risk of MSDs.

Definition 4: (Posture assessments): In ergonomics, there are some metrics used to determine the risk factor of a human posture, one of the most popular would be OWAS. Its goal is to evaluate posture and movement to assess their risk based on a set of guidelines. Its requires human input and observation but it does help as a method of validation for algorithms that try to perform similar solutions. For our case, the most useful metric would be the Rapid Upper Limb Assessment (RULA), which is used to estimate the risk of developing MSDs in the upper links, neck and trunk. It involves analyzing the worker's upper limb trunk[12].

B. Method

1) First Contribution. The model: A recurrent LSTM network will be used to realize the bad posture recognition system. This network is a type of neural network that can process data sequences and remember relevant information efficiently. The architecture is shown in figure 1.

Input and output size: The LSTM network will take sequences of body position landmarks as input. The input size will depend on the number of tracking points used to represent the body position at each time instant. For example, if a tracking point-based approach is used, one could have a 2D or 3D input for each point, resulting in a multi-dimensional input. The model output could be a binary value indicating whether the pose is good or bad. Multiple LSTM layers can be used to process sequences of input data and learn more complex and higher level features. For example, an architecture with two LSTM layers could be used, where the first layer processes the input sequence and the second layer processes the output of the first layer.

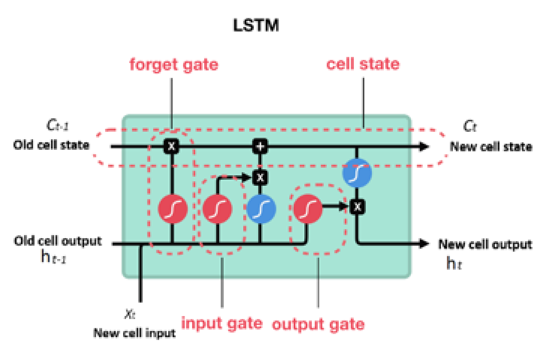

Fig. 1. Recurrent LSTM Network

LSTM layer size: The number of units in each LSTM layer is a hyperparameter to be adjusted during training. The size of the LSTM layer is related to the complexity of the model and its ability to learn more complex patterns. A layer with a larger number of units will have more capacity to learn patterns, but will also require more computational resources and may be prone to overfitting. In addition to the LSTM layers, other layers can be added to the model, such as activation layers, clustering layers, convolutional layers, etc. These layers can help extract more relevant features from the input and improve model performance.

Regularization: It is important to include regularization techniques in the model to avoid overfitting. This can be achieved by adding dropout layers, reducing the size of the layers, and increasing the amount of training data.

In summary, designing an LSTM network for bad posture estimation is a process that requires experimentation and tuning of the hyperparameters to find the architecture that best suits the data and the specific task. It is important to consider factors such as the size of the input and output, the number and size of LSTM layers, the inclusion of additional layers and regularization techniques to ensure optimal model performance.

2) Second Contribution. The dataset: The second contribution is a dataset to train LSTM models for the detection of bad posture. In our system, it is important to have a complete dataset with the necessary frames. This is a mixture of different datasets and images with desktop angles. First, a study had to be done on what are the most common postures of the workers, as well as the environment in which they work. Another important factor is the angle at which the camera is positioned, since the input we want to get is from the NAO robot, the angle of the images with which the model is trained with, has to be close to those that will be used with the NAO robot in the final system (which is shown on figure 2).

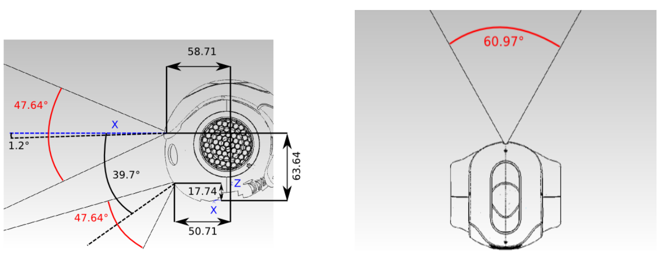

Fig. 2. Camera angles for the NAO Robot

3) Third Contribution. The pose estimation model: The system consists of two main components: the NAO robot and the LSTM-based pose estimation model. The NAO robot is a humanoid robot designed to interact with humans in various settings, such as education, entertainment, and healthcare. In our system, we utilized the NAO robot's built-in camera to capture images of the user's posture, which is a crucial input for the pose estimation model. The images are then fed into the LSTM-based pose estimation model, which is a type of recurrent neural network that is particularly effective in modeling sequential data, such as body joint positions over time. The LSTM-based model is trained on a dataset of labeled postures, including good posture and various types of bad posture, such as slouching, leaning, and hunching. The model learns to recognize different postures and detect bad posture by analyzing the temporal patterns of body joint positions in the input images. If the model detects bad posture, the NAO robot alerts the user with a voice message, which is designed to be polite and informative, asking them to adjust their posture. The voice message is an important feedback mechanism that helps the user correct their posture and prevent MSD. This flow can be seen in figure 3.

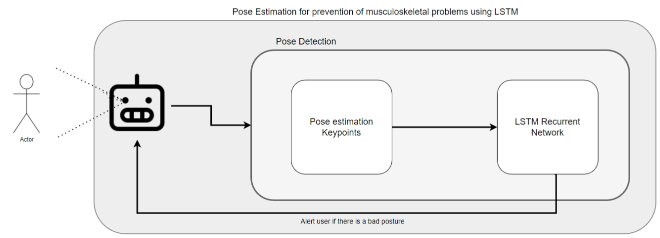

Fig. 3. System to prevent musclesqueletal problems

To integrate the LSTM-based pose estimation model with the NAO robot, we used the Python programming language and the NAOqi SDK, which is a software development kit that provides an interface to control the NAO robot's hardware and software components. We wrote a script that runs on the NAO robot and communicates with the pose estimation model running on a remote server. The script captures images from the NAO robot's camera and process them with the pose estimation method in order to get a sequence of landmarks that can then be feeded into the risk detection model. These set of landmarks must be 3D points. The methods we have tested for pose estimation are mainly two libraries. YoloV7[21], which is a which is a popular object detection model based on YOLO algorithm. This model has an implementation for human pose estimation that works really well. And the Media Pipe[20] open-source framework developed by Google. Which provides a lot of tools for AI including object and pose detection. The actual contribution, besides the usage of these available tools, would be the approach we're taking to improve their performance since we need them to work fast so the actual risk detection process doesn't take to long. We intend to enhance the performance of these algorithms by applying filters and preprocessing to the images as well as testing them in different aspect ratios, resolutions and fps. Also tweak the YoloV7 model a bit to make it focus on only one object in order to make it faster.

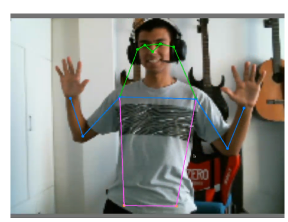

Fig. 4. Pose estimation example using yolov7

The prepossessed images are then fed into the LSTM-based pose estimation model, which outputs a binary classification result indicating whether the user is sitting in a good posture or a bad posture. If the model detects bad posture, the script triggers a voice message from the NAO robot asking the user to adjust their posture. The voice message is played through the NAO robot's built-in speaker, which is located in the head module, and is designed to be audible and clear, even in noisy environments.

IV. Experiments

In this section, we will provide a concise overview of the experiments conducted in our project, focusing on the comparison of pose estimation techniques between YoloV7 and MediaPipe. We will also discuss the implementation of an LSTM model for predicting bad posture. Additionally, we will outline the necessary requirements for replicating these experiments and provide a brief discussion of the obtained results.

The first experiment in our project involved comparing the performance of two pose estimation techniques: YoloV7 and MediaPipe. Pose estimation is a computer vision task that aims to detect and track human poses in images or videos. YoloV7 and MediaPipe are popular frameworks used for pose estimation, and we chose to evaluate their effectiveness in our research.

The second experiment focused on developing an LSTM (Long Short-Term Memory) model for predicting bad posture. This experiment aimed to address the problem of identifying and predicting incorrect body postures in real-time. The LSTM model was trained on a labeled dataset that consisted of time-series data capturing different body postures and corresponding labels indicating whether the posture was correct or incorrect. The experimental setup included a diverse dataset of images and videos containing various human sitting poses.

A. Experimental Protocol

In this subsection, we provide details about the environment on which we conducted the experiments. This includes information about the computer hardware we used and and some of the software we employed.

The following tests were executed on a laptop using an Intel i7-8750H (12) @ 2.200GHz, 16 GB of RAM running an Arch Linux system using Xorg Display Server. The source code of the code we display here can be found at https://github.com/Cesarmosqueira/Autoposture[13] there, a requirements.txt file shows the modules we used. Some of the most worth mentioning ones are Pytorch[16], Mediapipe[20], Naoqi[19], and OpenCV[15]

B. Pose Estimation

In the pose estimation section you will see the proposed experiments. In this case there are two: Frames per second and 3D landmarks. The aim of these experiments is to find the library that best balances performance with accuracy when displaying results.

1) FPS: The FPS evaluates the performance in execution of the library and its capacity to be fast. The more frames per second it has, the smoother the display and the more natural the user will be able to move.

For this we tested for a period of time to get a graph of FPS and calculate. Everything was worked under the same machine, as well as in the same environment and with the same person doing the same movements. We obtained the following results:

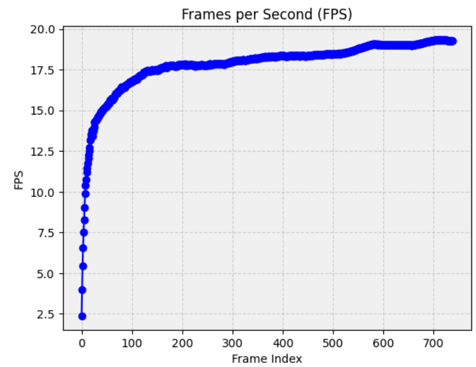

Fig. 5. Mediapipe FPS

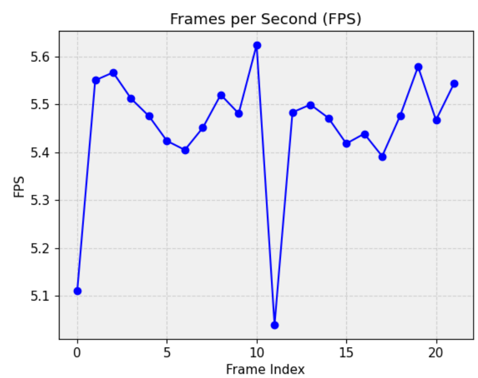

Fig. 6. YoloV7 FPS

You can see the difference in FPS and conscientiousness between Mediapipe5 and Yolo6 . The former shows up to 20 frames per second, which is a very promising figure for these systems that require a lot of real-time processing. Meanwhile, Yolo is around 5 frames per second in the same time. This result is well below what is expected for a library like yoloV7. It should be noted that it was tested using CPU using the source code provided YoloV7 repository[22].

If we compare the two models without making any enhancements, Mediapipe has a better image performance than yolo by a wide margin. As it depends on speed and is a key factor for our tool and system, this is an important factor to consider.

2) 3D Landmarks: The most important metric is the 3D landmarks of each library. These are the ones that will be sent to the lstm in order to perform the malposition prediction. For this purpose, pose estimation was performed using the same image, to compare and contrast the results.

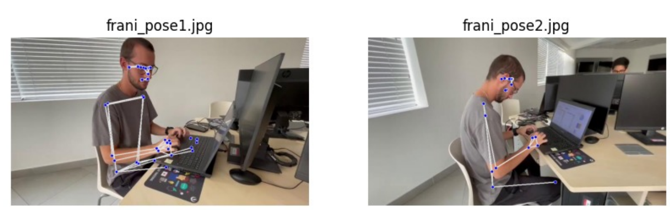

Fig. 7. Mediapipe: 3D landmarks

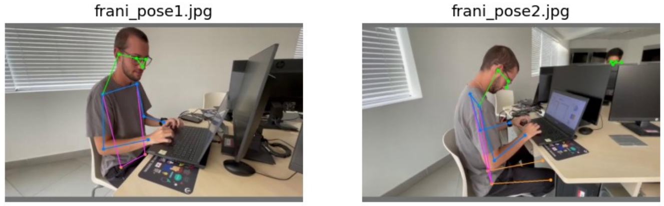

Fig. 8. YoloV7 3D landmarks

In Fig: 7 it can be seen how mediapipe doesn't estimate the pose properly, especially in the section of the hands and thighs. In addition, it does not have a visualization of the neck, which can be of great help for this project. In the first image of mediapipe it is possible to see clearly the torso, however in the second one it is more difficult to see and distinguish the points of the shoulder.

On the other hand in Fig: 8 we see that yoloV7 has clearer landmarks, with color distinctions between the main body parts for understanding. It also estimates the pose more accurately, with a landmark for the neck. Also, you can see how in the second image, the torso, hands, neck and thighs are still clearly visible, giving more detail.

3) Results: While mediapipe has much better results in Frames per second, giving a better fluidity and image quality. In the 3D landmarks section, it is not as accurate, not recognizing some body parts well. Meanwhile, Yolo is the opposite, resulting in worse frames per second but being able to better recognize the body in pose estimation.

References

[1] Jehangir Arshad et al. "An Intelligent Cost-Efficient System to Prevent the Improper Posture Hazards in Offices Using Machine Learning Algorithms". In: *Computational Intelligence and Neuroscience* 2022 (Aug. 2022). Ed. by Mohamed Abdelaziz, pp. 1–9. DOI: 10.1155/2022/7957148. URL: https://doi.org/10.1155/2022/7957148.

[2] Zhe Cao et al. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2018. DOI: 10.48550/ARXIV.1812.08008. URL: https://arxiv.org/abs/1812.08008.

[3] Hadi Daneshmandi et al. "Adverse Effects of Prolonged Sitting Behavior on the General Health of Office Workers". In: *Journal of Lifestyle Medicine* 7.2 (July 2017), pp. 69–75. DOI: 10.15280/jlm.2017.7.2.69. URL: https://doi.org/10.15280/jlm.2017.7.2.69.

[4] Firgan Feradov, Valentina Markova, and Todor Ganchev. "Automated Detection of Improper Sitting Postures in Computer Users Based on Motion Capture Sensors". In: *Computers* 11.7 (July 2022), p. 116. DOI: 10.3390/computers11070116. URL: https://doi.org/10.3390/computers11070116.

[5] Marta GÓMEZ-GALÁN et al. "Musculoskeletal disorders: OWAS review". In: *Industrial Health* 55.4 (2017), pp. 314–337. DOI: 10.2486/indhealth.2016-0191. URL: https://doi.org/10.2486/indhealth.2016-0191.

[6] Kaiming He et al. *Identity Mappings in Deep Residual Networks*. 2016. DOI: 10.48550/ARXIV.1603.05027. URL: https://arxiv.org/abs/1603.05027.

[7] Kaiming He et al. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: 10.1109/iccv.2017.322. URL: https://doi.org/10.1109/iccv.2017.322.

[8] Yongqiang Jiang et al. "WiLay". In: *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, Nov. 2019. DOI: 10.1145/3360774.3360812. URL: https://doi.org/10.1145/3360774.3360812.

[9] Alex Kendall, Matthew Grimes, and Roberto Cipolla. *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization*. 2015. DOI: 10.48550/ARXIV.1505.07427. URL: https://arxiv.org/abs/1505.07427.

[10] Audrius Kulikajevas, Rytis Maskeliunas, and Robertas Damaševičius. "Detection of sitting posture using hierarchical image composition and deep learning". In: *PeerJ Computer Science* 7 (Mar. 2021), e442. DOI: 10.7717/peerj-cs.442. URL: https://doi.org/10.7717/peerj-cs.442.

[11] Yu-Chi Lee and Ching-Hung Lee. "SEE: A proactive strategy-centric and deep learning-based ergonomic risk assessment system for risky posture recognition". In: *Advanced Engineering Informatics* 53 (Aug. 2022), p. 101717. DOI: 10.1016/j.aei.2022.101717. URL: https://doi.org/10.1016/j.aei.2022.101717.

[12] L. McAtamney and E. N. Corlett. "RULA: a survey method for the investigation of work-related upper limb disorders". In: *Applied ergonomics* 24.2 (1993), pp. 91–99.

[13] Cesar Mosqueira and Francesco Bassino. *Autoposture*. https://github.com/Cesarmosqueira/Autoposture. (2023).

[14] Taraneh Aminosharieh Najafi et al. "Development of a Smart Chair Sensors System and Classification of Sitting Postures with Deep Learning Algorithms". In: *Sensors* 22.15 (July 2022), p. 5585. DOI: 10.3390/s22155585. URL: https://doi.org/10.3390/s22155585.

[15] OpenCV Contributors. "OpenCV: Open Source Computer Vision Library". In: *Computing in Science & Engineering* 18.1 (2016), pp. 72–75. DOI: 10.1109/MCSE.2016.19.

[16] *PyTorch*. https://pytorch.org/. Year accessed (e.g., 2023).

[17] Martin Rajchl et al. *DeepCut: Object Segmentation from Bounding Box Annotations using Convolutional Neural Networks*. 2016. DOI: 10.48550/ARXIV.1605.07866. URL: https://arxiv.org/abs/1605.07866.

[18] Mark Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: (2018). DOI: 10.48550/ARXIV.1801.04381. URL: https://arxiv.org/abs/1801.04381.

[19] SoftBank Robotics. *NAOqi SDK*. https : / / www . softbankrobotics . com / emea / en / naoqi - developer - program/. 2023.

[20] Maria Vakalopoulou et al. "MediaPipe: A Framework for Building Perception Pipelines". In: *Proceedings of the 27th ACM International Conference on Multimedia*. ACM. 2019, pp. 2342–2345.

[21] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022. DOI: 10.48550/ARXIV.2207.02696. URL: https://arxiv.org/abs/2207.02696.

[22] WongKinYiu. *yolov7*. https://github.com/WongKinYiu/yolov7. year.

[23] Shichao Yue et al. "BodyCompass". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.2 (June 2020), pp. 1–25. DOI: 10.1145/3397311. URL: https://doi.org/10.1145/3397311.