sensors

MDPI

Article

A Deep-Learning Based Posture Detection System for Preventing Telework-Related Musculoskeletal Disorders

Enrique Piñero-Fuentes [1], Salvador Canas-Moreno [1], Antonio Rios-Navarro [1,2], Manuel Domínguez-Morales [1,2], José Luis Sevillano [1,2,*] and Alejandro Linares-Barranco [1,2]

[1] Architecture and Computer Technology Department, Escuela Técnica Superior de Ingeniería Informática—Escuela Politécnica Superior, University of Seville, 41012 Seville, Spain; epinerof@us.es (E.P.-F.); scanas@us.es (S.C.-M.); arios@us.es (A.R.-N.); mjdominguez@us.es (M.D.-M.); alinares@us.es (A.L.-B.)

[2] I3US: Research Institute of Computer Engineering, University of Seville, 41012 Seville, Spain

[*] Correspondence: jlsevillano@us.es

Abstract: The change from face-to-face work to teleworking caused by the pandemic has induced multiple workers to spend more time than usual in front of a computer; in addition, the sudden installation of workstations in homes means that not all of them meet the necessary characteristics for the worker to be able to position himself/herself comfortably with the correct posture in front of their computer. Furthermore, from the point of view of the medical personnel in charge of occupational risk prevention, an automated tool able to quantify the degree of incorrectness of a postural habit in a worker is needed. For this purpose, in this work, a system based on the postural detection of the worker is designed, implemented and tested, using a specialized hardware system that processes video in real time through convolutional neural networks. This system is capable of detecting the posture of the neck, shoulders and arms, providing recommendations to the worker in order to prevent possible health problems, due to poor posture. The results of the proposed system show that this video processing can be carried out in real time (up to 25 processed frames/sec) with a low power consumption (less than 10 watts) using specialized hardware, obtaining an accuracy of over 80% in terms of the pattern detected.

Keywords: convolutional neural network; skeleton; posture; telework; e-health

Citation: Piñero-Fuentes, E.; Canas-Moreno, S.; Rios-Navarro, A.; Domínguez-Morales, M.; Sevillano, J.L.; Linares-Barranco, A. A Deep-Learning Based Posture Detection System for Preventing Telework-Related Musculoskeletal Disorders. *Sensors* **2021**, *21*, 5236. https://doi.org/10.3390/s21155236

Academic Editor: Joanna Kolodziej

Received: 30 June 2021
Accepted: 30 July 2021
Published: 2 August 2021

1. Introduction

Year 2020 became a turning point worldwide, forcing the entire population to make changes in their daily habits due to the pandemic, in which we are still immersed. Among these changes, those that concern employment have a special repercussion: many jobs were lost in the last year, which was a tragedy for the world economy. However, even among the population that kept their jobs, there have been substantial changes in the way they work, leading to a notable increase in teleworking [1].

According to the Eurofund study performed in July 2020 [2], 33.7% of EU employees work at home full-time, while an additional 14.2% of EU employees work at home part-time. In 2018, less than 5% of EU employees were full-time teleworkers (and less than 10% were part-time teleworkers) [3]; therefore, the impact of the pandemic led to an increase of more than 650% among full-time teleworkers. This study also reveals that almost half of these employees have never teleworked before.

On the other hand, working with a computer for many hours a day can cause injuries mainly to the spine, but also to the shoulders, neck, arms and/or wrists; those injuries are usually caused by incorrect postures in front of the computer. These problems are documented in multiple studies [4–7] and reveal the need to maintain good postural hygiene to avoid these injuries. Moreover, the probability of adopting wrong postures increases when most of the latest teleworkers have never worked this way before.

We can define the posture as the position that each person adopts when performing his/her activities, when walking, sitting, standing or sleeping. Regarding the term postural hygiene, it can be defined as the set of rules that aims to maintain the correct position of the body, either in stillness or in movement, and thus avoiding possible injuries. A proper posture maintains symmetrical body alignment around the head–spine–pelvis axis with the total weight distributed equitably, while a bad posture does not respect this alignment. Among the factors that affect posture, we can enumerate some of them: genetic and anatomical factors, mood, tension, stress, sport, overweight, sedentary lifestyle, bad nutrition, physiological state of the musculoskeletal system, excessive muscle activity and, last but not least, the objects that are used or carried on a day-to-day basis (bed, sofa, bags, work chair, etc.).

So, according to the previous explanations, the continuous use of the work chair in front of a computer can provoke asymmetries in the body alignment that, finally, may lead to spinal injuries such as hyperlordosis (very marked curvature of the spine in the lower back), hyperkyphosis (increase in the anterior curvature of the spine vertebral and dorsal), scoliosis (structural malformation of the spine) or rectifications (reduction of normal curvature of the spine) [8]. That is why recommendations on postural hygiene when working in a chair are important to take into account [9].

To verify the correct spine posture while the user is sitting on a chair, the parameters that are usually used measure the angle at which the spine, shoulders and neck are positioned and moved. In this way, it is useful to measure certain parameters, such as the lateral back bend, back flexion/extension, shoulder adduction/abduction, lateral neck bend, and neck flexion/extension, among others [10]. The recommended angles for those parameters used to measure the correct postures have been detailed in ergonomic and biomechanic studies [11,12], obtaining four ranges of motion zones (0 to 3), according to the strain put on the joints. This range study is summarized in Table 1.

Table 1. Range (in degrees) of motion zones (0 to 3) for some of the parameters used to quantify the body posture.

Parameter	Range 0	Range 1	Range 2	Range 3
Back—Lateral bend	0–5	6–10	11–20	21+
Back—Flexion	0–10	11–25	26–45	46+
Back—Extension	0–5	6–10	11–20	21+
Shoulder—Alignment	0–5	6–10	11–20	21+
Shoulder—Arms adduction	0–5	6–12	13–24	25+
Shoulder—Arms abduction	0–13	14–34	35–67	68+
Neck—Lateral bend	0–5	6–12	13–24	25+
Neck—Flexion	0–9	10–22	23–45	46+
Neck—Extension	0–6	7–15	16–30	31+

As can be observed in Table 1, each parameter is divided into four ranges, according to the danger to the joints when maintaining those angles over a long period of time. Movements inside Zone 0 have a very low chance to damage the joints; these chances increase for Zone 1 (low), Zone 2 (medium) and Zone 3 (high). These ergonomic studies offer the possibility of developing recommendation systems in order to help the user to correct his/her posture; in fact, there are several works that obtained user's posture information by using a wide variety of sensors and systems [13–18]. However, only a few of them indicated recommendations to the user and, moreover, the information obtained was processed (or studied by the specialist) offline and do not provide any information to the user in real time [19–22].

On the other hand, to correct incorrect posture habits, the current procedures include an assessment by the specialist and an action plan based on physical exercises and adaptation of the workspace [23]. Although this process is effective, on many occasions, the specialist is consulted long after, when there is already some damage to the neck, shoulders or spine.

Because of that, the early detection of bad posture before they cause damage to the user is crucial. In this area, the occupational risk prevention services of most companies usually conduct courses for employees to adapt their posture habits in front of the computer. However, with the 2020 epidemiological crisis, there was a massive migration of workers from the company buildings to their homes to telework, so the control of the employees' workspace and posture habits was completely lost. This problem makes it more important to devise mechanisms capable of automatically detecting employees' posture and offering recommendations to avoid the health problems indicated above.

Regarding the tools and mechanisms used so far to develop recommendation systems and to prevent anomalies and/or physical problems, the use of Machine Learning (ML) and Deep Learning (DL) techniques has become very important. In recent years, these techniques can be found in multiple works related to the health field, both for medical image analysis (such as [24–26]) and to detect patterns through physiological sensors (such as [27–29]). Moreover, some of them are related to detecting the correct posture in certain circumstances to avoid joint pain and severe bone diseases (such as [30–32]).

In addition, there are some works, developed in recent years, that are focused on the detection of the user's skeleton through video processing [33,34]. These works usually use DL-based processing techniques with convolutional neural networks (CNN) to estimate the situation of the user's various joints. With these previous works, it is feasible to design a skeleton detection and tracking system using CNN and, with this information, to design a recommendation system, aiming for the proper posture of the worker in front of the computer. However, video processing using CNNs requires significant computational resources to provide a real-time response. It would be much more comfortable and transparent for the worker if the system was integrated in the monitor (both the vision sensor and the processing system).

Thus, in this work, we aim to design, develop and test a skeletal detection system using CNNs that subsequently estimates the user's posture and makes postural recommendations for the neck, shoulders and arms. This system is tested in real time, using various embedded systems, to measure both its effectiveness in detection and its response in real time as well as its energy consumption.

The rest of the manuscript is divided as follows: first, in the Materials and Methods Section, the characteristics of the posture detection system, the skeleton processing procedure and the system evaluation mechanism are presented. Next, the results after testing the system and evaluating the different hardware platforms are detailed and explained in the Results and Discussion Section. Finally, conclusions are presented.

2. Materials and Methods

The system is prepared to be placed on the computer screen or on a holder on top of it. The image acquisition device, a webcam or other type of camera, has to be pointing at the person to be able to capture them from the head to arms. This device is connected to an embedded system where the video is processed in two steps: first, the user's joints' positions are estimated, using a neural network classifier, and then the user's whole posture is processed and validated in order to provide recommendations attending to some of the parameters described in Table 1. In Figure 1, a global system overview is presented.

The most computationally expensive task in the proposed system is the execution of the neural network that estimates the user's joints. This type of algorithm needs sufficiently powerful hardware to run smoothly. Therefore, in this work, several embedded systems are evaluated in terms of execution time, power consumption and efficiency. The neural network used in this work requires supervised training, which means that it needs a training phase in which its internal parameters are adjusted to learn a specific task. This process is performed outside the embedded system due to its high computational workload, but the real-time classification is evaluated in the embedded system.

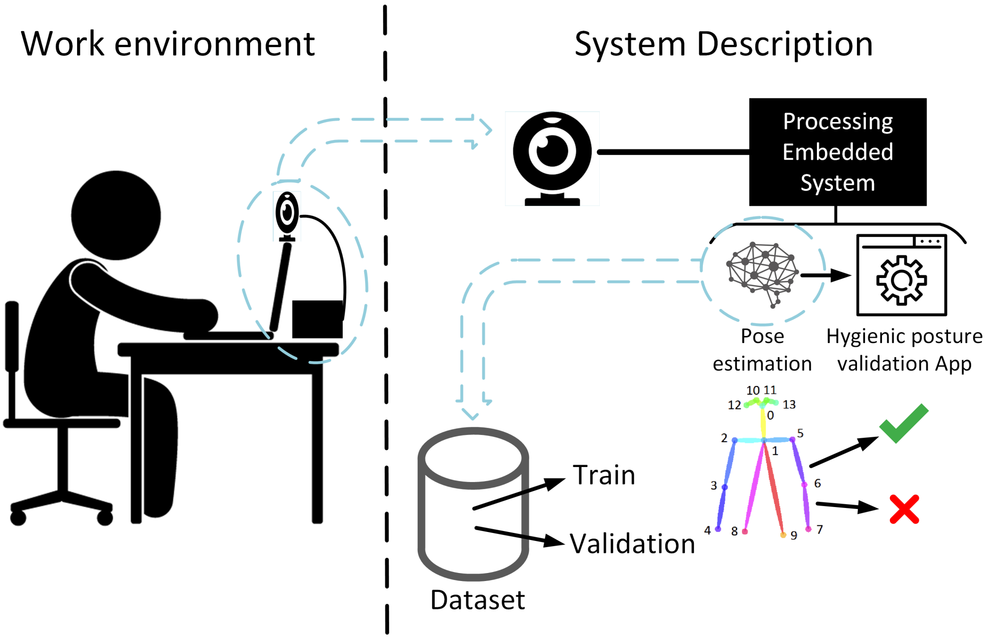


Figure 1. Application environment and global system overview.

Summarizing, the full system has two main blocks: the first one is the skeleton estimator used to obtain the joints' points; the second one is the recommendation system used to calculate the joints' angles, classify them inside the four-range space and notify the position to the user. The first subsystem (skeleton estimator) is based on a pre-trained model that is embedded in a specialized hardware. So the information used to train this subsystem is based on a third-part dataset (COCO dataset) that contains 66,808 images with, at least, one person each. This dataset contains people of both genders, and all kinds of races and ages. The second subsystem (recommendation system) uses the information obtained from the output of the skeleton estimation subsystem (points' list about the location of the joints) to calculate the different angles of the features used in this work and gives a notification to the user placed in front of the computer. The collected data for this work are used to test this second subsystem.

Next, the different parts of the processing system are described in depth. First, the embedded system that performs the posture detection process is detailed; secondly, the skeleton processing mechanism is presented; and, finally, the evaluation metrics used to test the system are shown.

2.1. Posture Detection System

To develop the posture detection system, a real-time pose estimation software called *TRT_Pose* was used [35]. This software is designed to run on NVIDIA devices, not only in desktop GPUs but also in embedded devices, that is, the NVIDIA Jetson family. So, this software is configured, executed and tested on different NVIDIA hardware devices that belong to that family. The main purpose of this evaluation is to obtain the performance metrics of those hardware boards in order to determine the best platform for this work. It is important to mention that, to obtain those evaluation metrics, the pose detection network was trained offline with the MSCOCO dataset [36], specifically on the *"Keypoints"* task (https://cocodataset.org/keypoints-2020 (last accessed on 20 June 21)). This public dataset contains 66,808 images with, at least, one person each. It contains people of both genders, and all kinds of races and ages; it is important to mention that the people tagged in this dataset are dressed in all kinds of attire (not just tight one), so bulkier garments are contemplated too, and the clothing factor does not affect the trained systems.

In Table 2, the hardware platforms used in this research are shown, detailing their computational performance, power consumption and other hardware characteristics and capacities. In this table, we can see the architectural differences of each device: from the most powerful one (NVIDIA GeForce GTX 1660 Super) to the less powerful (Jetson Nano). It is important to mention that the graphical board "NVIDIA GeForce GTX 1660 Super" is the only one that needs a desktop computer to be used (that is why columns "CPU" and "AI Performance" are not filled in Table 2); the rest of the hardware platforms are completely autonomous and do not need any other additional device to run. The main objective of using a classical GPU is to serve as a reference for the other devices.

As can be observed, the hardware devices are ordered from smallest (top) to largest (bottom) computing capacity, according to the GPU column. It is important to remark that, except for the first hardware device (Jetson Nano), the rest of them have more internal memory than the classic graphics card; this circumstance is due to the fact that, in a desktop computer, if the graphics card requires more memory, it usually uses the computer's memory. However, for this work, this characteristic does not affect the boards' performance.

Table 2. NVIDIA Jetson devices comparison. Data obtained from https://developer.nvidia.com/ (last accessed on 28 May 2021).

Device	GPU	CPU	Memory	Consumption (Min/Max)	AI Performance (TFLOPS)
Jetson Nano	128-core Maxwell GPU 921 MHz	Quad-Core ARM Cortex-A57 1.43 GHz	4 GB 64-bit LPDDR4 Dual Channel 1.6 GHz (25.6 GB/s)	5 W/10 W	0.472
Jetson TX2	256-core Pascal GPU 1.36 GHz	Dual-Core Denver2 64-Bit CPU and Quad-Core ARM Cortex-A57 2 GHz	8 GB 128-bit LPDDR4 1.866 GHz (59.7 GB/s)	7.5 W/15 W	1.33
Jetson Xavier NX	384-core Volta GPU 48 Tensor Cores 1.1 GHz	6-core Carmel ARMv8.2 64-bit CPU 1.9 GHz	8 GB 128-bit LPDDR4 x 1.6 GHz (51.2 GB/s)	10 W/15 W	21
Jetson AGX Xavier	512-core Volta GPU 64 Tensor Cores 1.4 GHz	8-core Carmel Armv8.2 64-bit CPU 2.3 GHz	32 GB 256-bit LPDDR4 x 2.133 GHz (136.5 GB/s)	10 W/30 W	32
NVIDIA GeForce GTX 1660 Super	1408-Core Turin GPU 1.785 GHz	N/A	6 GB GDDR6 192-bit 14 Gbps (336 GB/s)	10 W/125 W	N/A

Once the hardware devices used for the performance test are presented, the protocol used for capturing the user's pose and obtaining the video streams used to test the system is detailed. As there is no available dataset that contains the information needed to this work (to the authors' knowledge) we collected around 80 videoclips from 5 different users performing working tasks in front of a computer.

To capture the frames from each video stream, a webcam was placed on the computer's screen frame, calibrated for each user at the same height as the line of sight. Each user was sitting on a height-configurable chair in front of the computer (so a full vision of the upper side of the human body is obtained) and he/she was instructed to compose a few lines in a text document. As we needed incorrect user's poses to test our system, each user was ordered to perform the actions in different ways: first, those actions were performed in his/her natural pose; second, the user was forced to adopt an incorrect neck pose; third, the user was forced to adopt an incorrect shoulder balance; and, fourth, the user was forced to adopt an incorrect arm elevation pose.

The collected dataset was analyzed one by one by selecting several frames from each video clip to obtain the ground truth about the user pose adopted in each video. So, the entire dataset was divided into three subsets: neck poses, shoulder poses and arm poses. Additionally, each subset was labeled from range 0 to range 3, according to the degrees of variation adopted by the user on each video clip (the thresholds are shown in Table 1), which was analyzed offline. Some captures about the system setup when capturing the dataset's video clips can be observed in Figure 2.

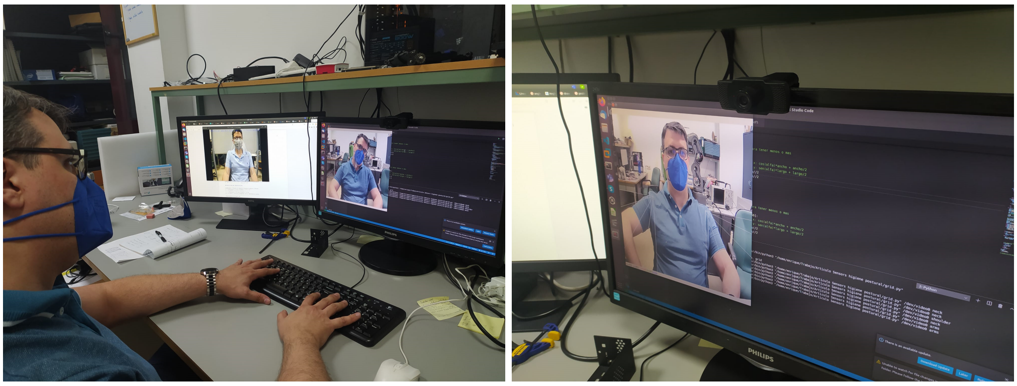


Figure 2. Setup with the webcam on the screen monitor.

The final classification results of the system corresponds to the percentage of coincidence of the ranges estimated by the system in real time with respect to the established labels. Next, the skeleton processing stage, which allows to obtain the user's joint points in real time, is described.

2.2. Skeleton Processing

As mentioned in the previous section, for the real-time skeleton detection in this work, we used the software called *TRT_Pose*. Specifically, we used a modified version of the original repository, which is called *TRT_Pose_Demo* [37].

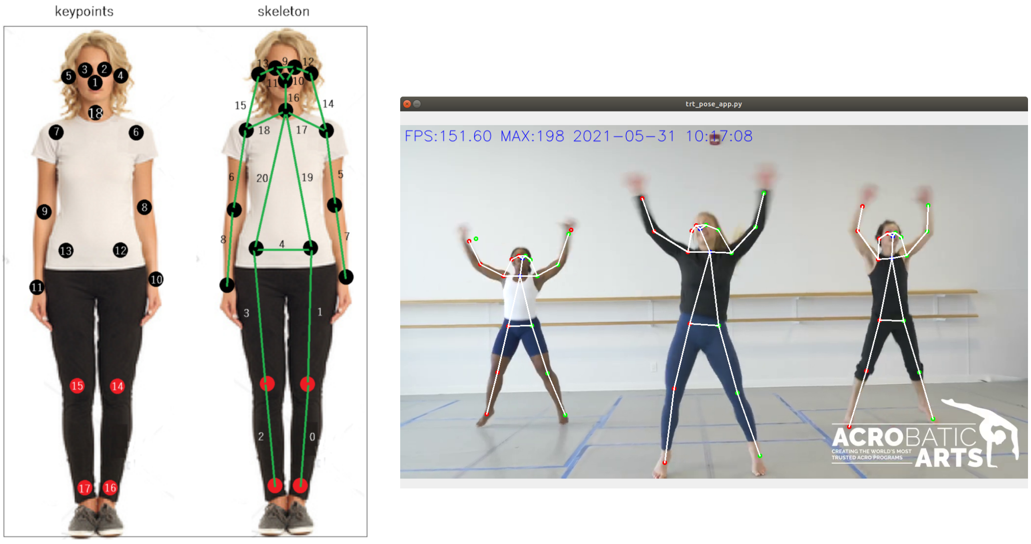
The processing framework is based on a CNN model implemented in Pytorch [38], whose name is *ResNet* [39]. More concretely, we used a modified version of that model called *ResNet18* [40,41]. The benefits of using a "ResNet" arise with an idea similar to long short-term memory (LSTM) networks (recurrent neural networks) in which the temporal factor is taken into account to perform the classification. "ResNet" makes use of internal connections between layers that are not contiguous to communicate information from later (or earlier) moments in time with a specific moment. In this way, to perform the classification, not only can the information of the current frame be taken into account, but also the information of subsequent and/or previous frames (using connections similar to the bypasses that can be seen in the internal pipeline of a processor). About the variant *ResNet18*, the end number indicates the number of layers of the model.

The final result of the deep learning model is a set of points related to the user joints detected, being able to detect more than one user in each frame. For this model, an user skeleton is defined by 18 points (this number is not related to the *ResNet* variant, as it is just a coincidence); the location of these points determines the person's adopted pose. A global vision of the 18 points that this model detects, and their locations placed over the original image, is shown in Figure 3a.

Figure 3b shows an example of the real-time execution of the skeleton estimation system. As can be seen in both figures, the shape or pose of a person is delimited by the 18 points indicated above connected to each other in a specific way.

Once the skeleton estimation model task is defined, it should be noted that, although the network is trained and defined in PyTorch, a series of optimizations are performed prior to the execution of the model in real-time hardware (pre-processing step) for the new NVIDIA platforms that have tensor cores. So, the network previously passed through an optimization process that converts it for its execution on those specialized hardware. With these optimizations, besides allowing to be executed in that hardware, the data representation model is changed from a 32-bit Floating Point to a 16-bit Floating Point number representation.

Those optimizations are included because it is intended that the execution in the different embedded systems is as efficient as possible, always looking for the lowest possible consumption.



FPS:151.60 MAX:198 2021-05-31 10:17:08

(**a**) (**b**)

Figure 3. Output of the first subsystem: (**a**) structure and location of the skeleton's keypoints that represent the joints; (**b**) execution example.

The performed pre-processing consists of converting the model trained in PyTorch format to a format compatible with NVIDIA's *Tensor RT* [42] (TRT) library, which is designed with the aim of accelerating the neural network estimation process. For this purpose, a tool called *Torch2TRT* [43] was used, which precisely converts the trained PyTorch model into a version compatible with the Tensor RT libraries. The number precision reduction from 32-bit floating point to half precision (Float16) during optimization reduces the computation time as well as energy consumption, at the cost of reducing the final accuracy of the model by less than 2%, according to previous studies [44]. This is a technique widely used when tuning models for execution in embedded and/or edge-computing systems since, usually, the loss of accuracy is minimal compared to the gain in performance.

Finally, in the last subsection, the processing and metrics performed to evaluate the implemented system are detailed.

2.3. System Evaluation

In this section, we detail how our system recognizes a correct posture, how we are going to evaluate the ground-truth postures collected and the metrics used to evaluate the performance of the system.

2.3.1. Posture Recognition

Given a posture, there are multiple ways in which we can assess whether or not it is correct. This ultimately means that there are a lot of different factors that apply, as we can see in [10] and in Table 1, so evaluating a specific posture is a non-trivial task and can be solved using different methods and metrics. To keep things simple while at the same time being able to solve the problems related to the current teleworking needs, we have decided to analyze three aspects of the posture: the neck lateral bend, the shoulders alignment and the arms abduction (also related to the shoulders). These parameters can be seen marked in Figure 4. Each of these parameters has four possible zones in which they can be, ranging from good posture (healthy) to bad posture (very unhealthy) progressively. The first zone, denoted as "Range 0" in Table 1, corresponds to the best posture and therefore the healthiest, while the fourth zone, "Range 3" in the table, corresponds to the worst possible posture. With this in mind, we take into account the already mentioned parameters for each image evaluated and classify them using Table 1's criteria.

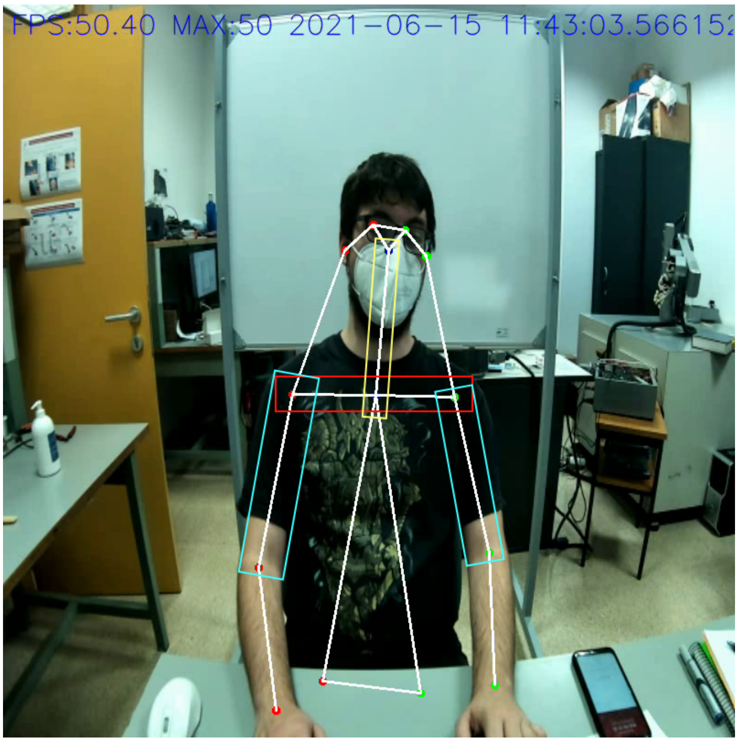


Figure 4. Evaluable parameters from test dataset. Parameters are marked over the CNN output skeleton with different colors: neck lateral bend (yellow), shoulder alignment (red) and arms abduction (cyan).

2.3.2. Posture Evaluation

Once we know how to recognize a good posture, we need to apply it to the postures collected during the work. For that, we can take advantage of the skeleton processing system's output to evaluate the angle of rotation of each evaluable parameter and classify it as described before. In order to do this, that output needs to consist not only of a video with the estimated skeleton, but also a collection of data that gives details about which keypoints are estimated at each step of time. Given this information, we can relate the timestamp of a specific frame with the timestamp of the data collection described earlier to obtain the pertinent keypoints and perform the required calculations. This is an offline procedure performed to manually tag the dataset collected. In this work, the accuracy of the collected dataset is evaluated, performing a proof of concept of the final system. After evaluating it, in future works, it will be adapted to a real-time application.

Calculations are performed, taking into account the following: we only have two points per parameter and we need to calculate the angle of the line expressed by those two points to classify it inside the mobility range (ranges detailed in Table 1). The points used for evaluating each parameter are detailed in Table 3.

Table 3. Points used to evaluate each parameter.

Parameter	Point 1 (x)	Point 2 (y)
Shoulder—Alignment	Left Shoulder	Right Shoulder
Right Arm adduction	Right Shoulder	Right Elbow
Left Arm abduction	Left Shoulder	Left Elbow
Neck—Lateral bend	Nose	Neck

These two points form the part of the skeleton that was estimated by the CNN and are taken from the data collection mentioned before, while the angle calculation is performed with the *atan2* function. This function is evaluated as shown in Equation (1).

$$atan2(y,x) = \begin{cases} arctan(\frac{y}{x}) & \text{if } x > 0 \\ arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases} \tag{1}$$

2.3.3. Metrics

The main goal of the system is to classify a series of angles extracted from the keypoints shown in Table 3 estimated by the CNN. So, in order to evaluate the results obtained after the estimations, we compare the tag stored during the recording process with the results obtained after obtaining the mobility ranges extracted with the degrees resulting from Equation (1). Therefore, we may define the accuracy of the system, A, as the ratio between the correct predicted outputs and the total predicted outputs per zone or class, denoted by Equation (2). TP and TN mean *true positives* and *true negatives*, while FP and FN mean *false positives* and *false negatives*. These parameters compose the output of the system, and tell us how the system predicted each and every of its inputs, whether they were correct or not. For each class, the information concerning the classification results is extracted by looking at that class independently. Another way is to see the global ranking only, but that provides much less information.

$$A_{total} = \frac{TP + TN}{TOTAL} \quad , where: \quad TOTAL = TP + TN + FP + FN \tag{2}$$

After presenting the implemented system and how it will be evaluated, in the next section, the results obtained by the pose detection system are detailed.

3. Results and Discussion

In this section, we detail the dataset collected and the performed results of the postural study, as well as a comparison with other related studies.

3.1. Collected Data

For this work, and after a second data collection process, 12 people have participated to record the dataset used. A series of recordings were performed for each person, in which each one held a different posture for each recording, so that we may have a variety of data to process and classify. Finally, a total amount of 151 correct recordings were made, and a total of 148 frames were processed (some of them were discarded due to recording errors).

The final dataset contains information about 12 users in their early 20's to early 50's, from both genders and both dominant hands.

The aim of this work is to evaluate the system, which is why random frames were extracted from the video recordings instead of evaluating the whole videos (this way, the tag process and the evaluation are more accurate); however, the system can work in real time, processing frame by frame.

3.2. Results

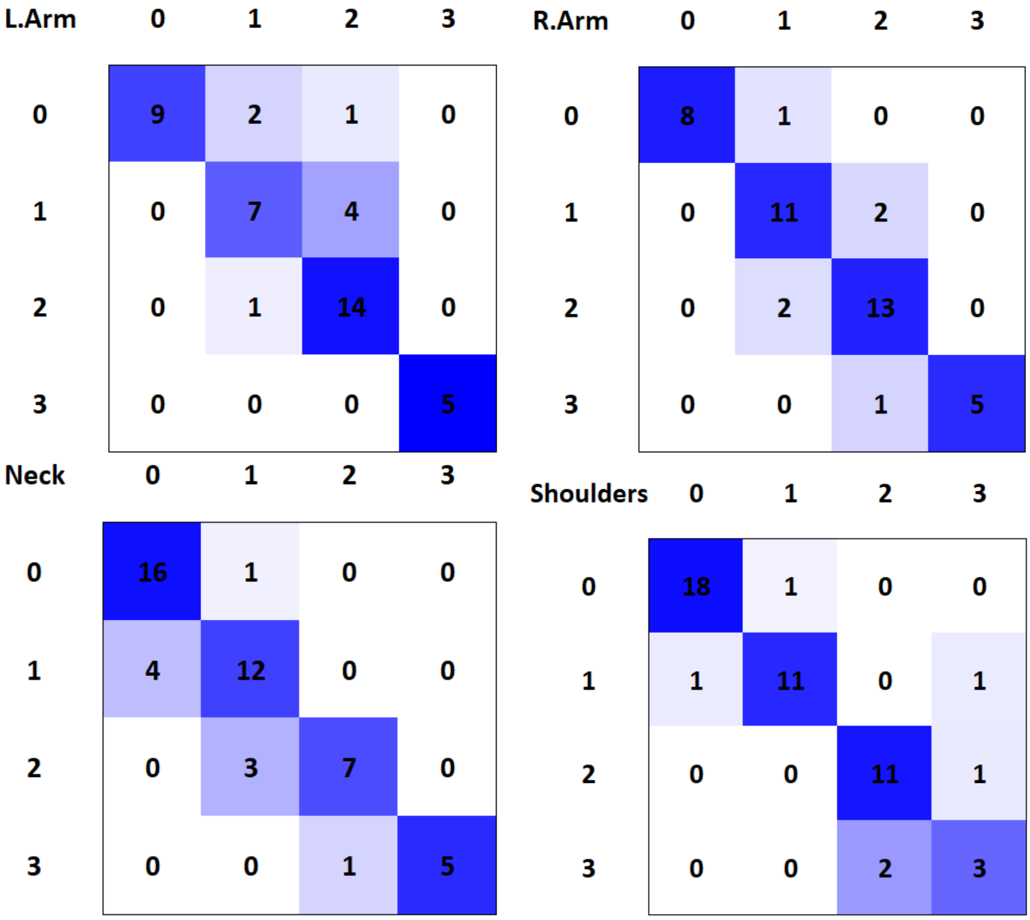
The system achieved a total accuracy of 84.1%, as can be seen in Table 4. As can be observed, the accuracy of each parameter is similar, obtaining the worst result (81.4%) with "Left Arm Abduction" and the best result (87.7%) with "Shoulder Alignment". It is important to remark that the dataset includes subjects of both genders, ages from early 20's to early 50's, both dominant hands and different clothes.

Table 4. Accuracy results for each parameter.

	Right Arm	Left Arm	Neck	Shoulder	Mean
Accuracy	0.860	0.814	0.816	0.877	0.841

The system's predictions per class can be visualized in Figure 5, which are the confusion matrices for each class. In them, the proportion of predictions against ground truth values is displayed, being that the main diagonal of the matrix is the correct predictions of the system. With these matrices, TP, TN, FP and FN values can be obtained per class, while also having a compact representation.

It can be observed that the best results, as shown in Table 4, are presented for the "Shoulder Alignment" parameter (Figure 5-top-right). On the other hand, we can observe two peculiar cases: one of them in "Left Arm Abduction", where a range 0 case is classified as range 2; and the other one in "Shoulder Alignment" parameter, where a range 1 case is classified as range 3. Apart from those two errors, the rest of the errors occur from one range to a contiguous one.



L.Arm	0	1	2	3
0	9	2	1	0
1	0	7	4	0
2	0	1	14	0
3	0	0	0	5

R.Arm	0	1	2	3
0	8	1	0	0
1	0	11	2	0
2	0	2	13	0
3	0	0	1	5

Neck	0	1	2	3
0	16	1	0	0
1	4	12	0	0
2	0	3	7	0
3	0	0	1	5

Shoulders	0	1	2	3
0	18	1	0	0
1	1	11	0	1
2	0	0	11	1
3	0	0	2	3

Figure 5. Confusion matrices for right arm abduction (**top left**) left arm abduction (**top right**) neck lateral bend (**bottom left**) and shoulder alignment (**bottom right**).

Moreover, knowing that the dangerous ranges are 2 and 3, a critical error would be produced if a dangerous pose (range 2 or 3) is classified as a safe pose (range 0 or 1). Evaluating only those two classes (critical and safe) for each parameter, the accuracy is improved. The results obtained after this group can be observed in Table 5.

Table 5. Accuracy results for each parameter using a 2-class classifier: safe (range 0–1) and dangerous (range 2–3).

	Right Arm	**Left Arm**	**Neck**	**Shoulder**	**Mean**
Accuracy	0.906	0.860	0.938	0.979	0.924

As can be observed in Table 5, the mean accuracy in this case is 92.4%. The worst case is shown with "Left Arm Abduction", obtaining a 86% accuracy; the best case is obtained for "Shoulder Alignment" with a 97.9% accuracy.

The dataset collected is not very large, but there is a great deal of variability among the participants. For this reason, an exhaustive study of the testing dataset was carried out, using bootstrap re-sampling. In this case, we have generated 15 random testing subsets, each one of them taking 50% of the samples from the full dataset. Each of these 15 subsets was evaluated independently, obtaining the classification accuracy for each of the parameters studied (neck, shoulders, right arm and left arm) and testing the 4-classes classification (range 0–3) and the 2-classes classification (ranges 0–1 and 2–3). The results obtained can be seen in Table 6.

Table 6. Bootstrap re-sampling evaluation with 15 random subsets.

	4 Classes				**2 Classes**			
Set	**Neck**	**Shoulder**	**R.Arm**	**L.Arm**	**Neck**	**Shoulder**	**R.Arm**	**L.Arm**
1	0.909	0.909	0.917	0.70	1	1	0.917	0.70
2	0.826	0.909	0.857	0.809	0.956	1	0.905	0.857
3	0.818	1	0.809	0.773	0.864	1	0.905	0.864
4	0.955	0.955	0.952	0.857	1	1	0.952	0.857
5	0.864	0.955	0.809	0.857	0.955	1	0.905	0.905
6	0.955	1	0.864	0.727	1	1	0.909	0.818
7	0.864	0.955	0.773	0.773	0.955	1	0.864	0.818
8	0.955	0.909	0.952	0.809	1	1	0.952	0.857
9	0.818	0.955	0.809	0.857	0.955	1	0.905	0.905
10	1	0.909	0.864	0.762	1	1	0.864	0.809
11	0.864	0.955	0.818	0.818	0.955	1	0.909	0.909
12	0.864	0.955	0.809	0.773	1	1	0.857	0.909
13	0.909	1	0.773	0.773	1	1	0.818	0.818
14	0.818	0.955	0.809	0.857	0.955	1	0.857	0.905
15	0.864	1	0.809	0.818	0.955	1	0.905	0.864
MIN	0.818	0.909	0.773	0.70	0.864	1	0.818	0.70
MAX	1	1	0.917	0.857	1	1	0.952	0.909
MEAN	0.885	0.955	0.842	0.797	0.969	1	0.895	0.853
STD	0.049	0.024	0.047	0.041	0.028	0	0.028	0.040
Improve					9.54%	4.76%	6.31%	6.94%

As can be seen in Table 6, the evaluation by re-sampling the dataset confirms some of the points previously indicated in the overall evaluation. On the one hand, the parameter for which the worst accuracy is obtained is the left arm, although, in this evaluation, a greater closeness with the right arm is observed (difference of less than 5%), which means that the detection of the position of the arms seems to be more complex than the detection of the remaining parameters. Secondly, it is confirmed that the most accurate parameter is the shoulders, obtaining a significant difference with respect to the others. Finally, neck detection is the second parameter in terms of classification perception. In short, qualitatively, it is similar to the global results but with slightly varying classification percentages (which depends on the random sample that was classified in each case).

With respect to the standard deviation after the classification of the 15 subsets, it can be seen that the shoulders are the ones with the lowest deviation, while the remaining parameters have a similar standard deviation.

If we consider the second part of Table 6, where results regarding the classification for the 2-class case are presented instead of the 4-class case, significant improvements can be observed. However, the relative accuracy remains the same, with the shoulders being the best ranked parameter, the neck the second-best ranked parameter and, in last position, the arms (with a slight variation between the right and the left arm, which is the one with the worst accuracy). As for the standard deviation, they are significantly reduced, except in the case of the left arm, which maintains practically the same value.

Therefore, with these data, we can assume that the complete dataset has consistent results and that all the samples follow the same pattern by having a standard deviation in all cases lower than 5%. Likewise, the fact that the classification with two classes yields much better results than those obtained with four classes is corroborated.

3.3. Performance Tests

A performance study on the different hardware platforms presented was also conducted in parallel. In it, we measured the consumption and total throughput of every device while running the CNN, so that we may give at least a recommendation about which device would be a good fit for setting up a system, such as that presented, taking into account the already mentioned parameters. The results can be seen in Figures 6–8.

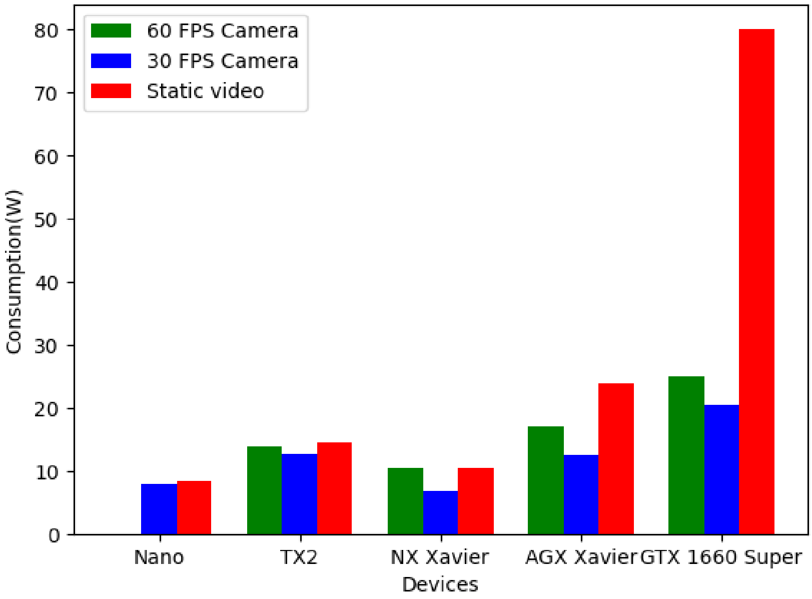


Figure 6. Consumption in watts for each device and test.

To be able to measure such parameters, three tests were designed. In two of them, the CNN's input is a real-time camera but one has a frame rate of 30 FPS and the other one has 60 FPS. The third test consists of the classifying of an offline video, which was made using only one single repeated frame; this way, the memory and other aspects would not be a bottleneck and so we could obtain a significant reference value of maximum throughput.

As we can see, almost all the devices presented would be a good fit; however, given the obtained results, a viable recommendation for building a system like the one presented would be the NX Xavier, as its efficiency is the maximum achieved, with 7 frames processed per watt consumed in the offline video test and 6 for the 60 FPS camera test.

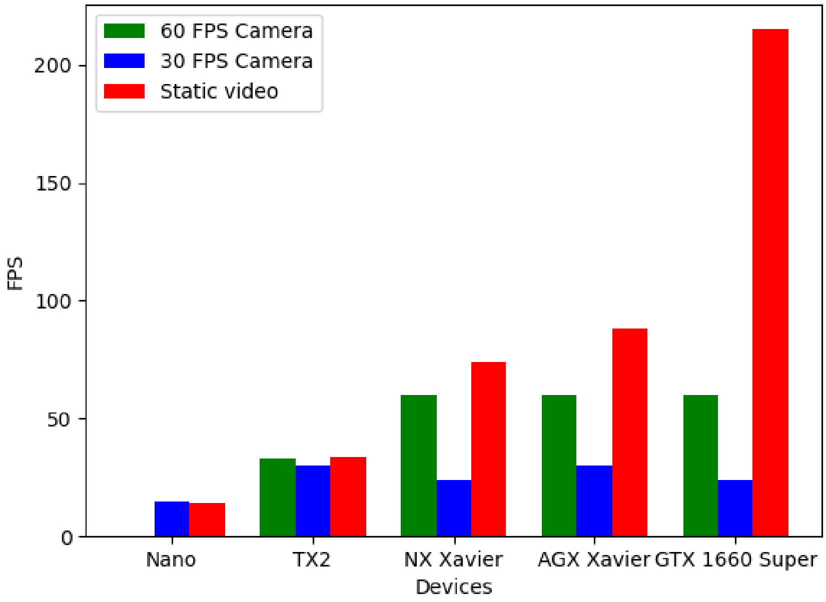


Figure 7. Achieved frame rates per device and test.

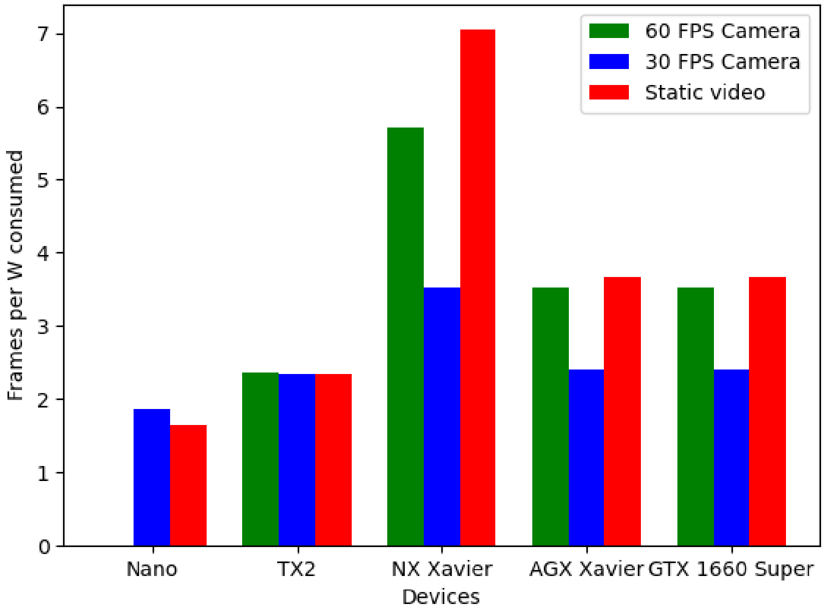


Figure 8. Ratio frame processed per watt consumed for each device and test.

4. Conclusions

Due to the pandemic situation of year 2020, the number of teleworkers has increased significantly. Such an amount of hours in front of a computer in a work space not properly adapted for that workload can lead to postural problems for the employee. That is why a system like the one proposed in this work is necessary.

After evaluating several hardware platforms and testing the system with multiple users, the results obtained in this study show that a postural recommendation system is more than viable, using the resources presented. Results showed a posture detection accuracy over 80% for the 4-class original problem, and more than 90% for the 2-class classification system. On the other hand, the hardware platforms tested allows the system to run in a real-time environment with low power consumption requirements. With these two points in favor, we can conclude that the system can work completely autonomously and without the intervention of a computer, providing information in real time.

Although this work was implemented and tested in a laboratory environment with a not very extensive dataset, the results are very promising. Moreover, this only reinforces the fact that there is still quite a lot to gain from automating and detailing the process. Future research may include real-time video processing with an automatic recommendation system, a collection of incorrect poses during real work sessions to improve overall postures, or even changes to the CNN used in order to obtain a more detailed pattern recognition that would allow for the checking of other parameters, such as the spine, which requires more than a single straight segment to be evaluated.

The metrics obtained in the results should be taken as further confirmation of the hypothesis, as the system is able to recognize correctly most of the inputs it is fed. A bigger study with more people and/or parameters is also interesting, as it would serve as even more proof that this kind of work is viable.

Author Contributions: Conceptualization, A.L.-B. and J.L.S.; methodology, E.P.-F. and M.D.-M.; software, E.P.-F.; validation, S.C.-M., A.R.-N. and A.L.-B.; formal analysis, A.R.-N., M.D.-M. and J.L.S.; investigation, E.P.-F., A.R.-N. and M.D.-M..; data curation, E.P.-F. and S.C.-M.; writing—original draft preparation, E.P.-F., S.C.-M., A.R.-N. and M.D.-M.; writing—review and editing, A.R.-N. and M.D.-M., J.L.S. and A.L.-B.; supervision, A.R.-N. and Linares-Barranco. A..; project administration, A.L.-B.; funding acquisition, J.L.S. and A.L.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Spanish Agencia Estatal de Investigación (AEI) project MINDROB: "Percepción y Cognición Neuromórfica para Actuación Robótica de Alta Velocidad (PID2019-105556GB-C33/ AEI/10.13039/501100011033)" and by "Fondo Europeo de Desarrollo Regional" (FEDER) and by "Consejería de Economía, Conocimiento, Empresas y Universidad" of the Junta de Andalucía, under Programa Operativo FEDER 2014-2020 (Project US-1263715).

Institutional Review Board Statement: Ethical review and approval were waived for this study, because all the participants belong to the research group that develops the work and the test were performed in a laboratory environment.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Acknowledgments: The authors would like to thank the Spanish Agencia Estatal de Investigación (AEI) for supporting this work. The work of Enrique Pinero-Fuentes was supported by a Formación de Personal Investigador Scholarship from the Spanish Ministry of Education, Culture and Sport.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Milasi, S.; González-Vázquez, I.; Fernández-Macías, E. *Telework in the EU before and after the COVID-19: Where We Were, Where We Head to*; Science for Policy Brief; European Commission: Brussels, Belgium, 2020.
2. Bejaković, P. Eurofound: Living, Working and Covid-19, Covid-19 Series. *Rev. Soc. Polit.* **2021**, *28*, 115–117.
3. Eurostat. How Usual Is It to Work from Home? 2020. Available online: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/DDN-20200424-1 (accessed on 30 May 2021).
4. Larrea-Araujo, C.; Ayala-Granja, J.; Vinueza-Cabezas, A.; Acosta-Vargas, P. Ergonomic Risk Factors of Teleworking in Ecuador during the COVID-19 Pandemic: A Cross-Sectional Study. *Int. J. Environ. Res. Public Health* **2021**, *18*, 5063. [CrossRef] [PubMed]
5. Moro, A. *Impact of the COVID-19 Confinement Measures on Telework in Italy-A Qualitative Survey*; Technical Report; JRC Working Papers Series on Labour; Education and Technology: Econstor, Italy, 2020
6. Weaver, S.J.; Champion, M. Are laptop computers a health risk for an aging population? *Gerontechnology* **2010**, *9*, 415–420
7. Nag, P.K. Musculoskeletal disorders: Office menace. In *Office Buildings*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 105–126.
8. Ventura, C.; Denton, E. Head, Neck, and Spinal Injuries. In *The Emergency Medical Responder*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 41–49.
9. Ohlendorf, D.; Sosnov, P.; Keller, J.; Wanke, E.M.; Oremek, G.; Ackermann, H.; Groneberg, D.A. Standard reference values of the upper body posture in healthy middle-aged female adults in Germany. *Sci. Rep.* **2021**, *11*, 1–10. [CrossRef] [PubMed]
10. Openshaw, S.; Taylor, E. *Ergonomics and Design a Reference Guide*; Allsteel Inc.: Muscatine, IA, USA, 2006.
11. Chaffin, D.B.; Andersson, G.B.; Martin, B.J. *Occupational Biomechanics*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
12. Woodson, W.E.; Tillman, B.; Tillman, P. *Human Factors Design Handbook: Information and Guidelines for the Design of Systems, Facilities, Equipment, and Products for Human Use*; McGraw-Hill: New York, NY, USA, 1992

13. Mallare, J.C.T.; Pineda, D.F.G.; Trinidad, G.M.; Serafica, R.D.; Villanueva, J.B.K.; Cruz, A.R.D.; Vicerra, R.R.P.; Serrano, K.K.D.; Roxas, E.A. Sitting posture assessment using computer vision. In Proceedings of the 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, Philippines, 1–3 December 2017; pp. 1–5.
14. Van Niekerk, S.M.; Louw, Q.; Vaughan, C.; Grimmer-Somers, K.; Schreve, K. Photographic measurement of upper-body sitting posture of high school students: A reliability and validity study. *BMC Musculoskelet. Disord.* **2008**, *9*, 1–11. [CrossRef] [PubMed]
15. Khandelwal, I.; Stollenwerk, K.; Krüger, B.; Weber, A. Posture Classification Based on a Spine Shape Monitoring System. In Proceedings of the International Conference on Computational Science and Its Applications, Saint Petersburg, Russia, 1–4 July 2019; pp. 494–509.
16. Gal-Nadasan, N.; Stoicu-Tivadar, V.; Poenaru, D.V.; Popa-Andrei, D.; Gal-Nadasan, E. Computer Aided Posture Screening Using the Microsoft Kinect on Professional Computer Users for Malicious Postures. In Proceedings of the International Workshop Soft Computing Applications, Arad, Romania, 24–26 August 2016; pp. 298–307.
17. Bibbo, D.; Carli, M.; Conforto, S.; Battisti, F. A sitting posture monitoring instrument to assess different levels of cognitive engagement. *Sensors* **2019**, *19*, 455. [CrossRef] [PubMed]
18. Kulikajevas, A.; Maskeliunas, R.; Damaševičius, R. Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Comput. Sci.* **2021**, *7*, e442. [CrossRef] [PubMed]
19. El-mir, Y.; Ivarsson, J. Ergonomic Posture Correction through a Camera Live Feed and Its Applicability in Terms of Usability. Available online: https://www.uppsatser.se/uppsats/4750f68d06/ (accessed on 30 May 2021)
20. Muppavram, S.; Patel, N.; Nadeem, M. Posture Alert. In Proceedings of the 2018 IEEE Region Ten Symposium (Tensymp), Sydney, NSW, Australia, 4–6 July 2018; pp. 213–218.
21. Chopra, S.; Kumar, M.; Sood, S. Wearable posture detection and alert system. In Proceedings of the 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 25–27 November 2016; pp. 130–134.
22. Haller, M.; Richter, C.; Brandl, P.; Gross, S.; Schossleitner, G.; Schrempf, A.; Nii, H.; Sugimoto, M.; Inami, M. Finding the right way for interrupting people improving their sitting posture. In Proceedings of the IFIP Conference on Human-Computer Interaction, Lisbon, Portugal, 5–9 September 2011; pp. 1–17.
23. Falla, D.; Jull, G.; Russell, T.; Vicenzino, B.; Hodges, P. Effect of neck exercise on sitting posture in patients with chronic neck pain. *Phys. Ther.* **2007**, *87*, 408–417. [CrossRef] [PubMed]
24. Duran-Lopez, L.; Dominguez-Morales, J.P.; Corral-Jaime, J.; Vicente-Diaz, S.; Linares-Barranco, A. COVID-XNet: A custom deep learning system to diagnose and locate COVID-19 in chest X-ray images. *Appl. Sci.* **2020**, *10*, 5683. [CrossRef]
25. Civit-Masot, J.; Domínguez-Morales, M.J.; Vicente-Díaz, S.; Civit, A. Dual machine-learning system to aid glaucoma diagnosis using disc and cup feature extraction. *IEEE Access* **2020**, *8*, 127519–127529. [CrossRef]
26. Ma, X.; Niu, Y.; Gu, L.; Wang, Y.; Zhao, Y.; Bailey, J.; Lu, F. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognit.* **2021**, *110*, 107332. [CrossRef]
27. Muñoz-Saavedra, L.; Luna-Perejón, F.; Civit-Masot, J.; Miró-Amarante, L.; Civit, A.; Domínguez-Morales, M. Affective state assistant for helping users with cognition disabilities using neural networks. *Electronics* **2020**, *9*, 1843. [CrossRef]
28. Sharma, M.; Dhiman, H.S.; Acharya, U.R. Automatic identification of insomnia using optimal antisymmetric biorthogonal wavelet filter bank with ECG signals. *Comput. Biol. Med.* **2021**, *131*, 104246. [CrossRef] [PubMed]
29. Tahura, S.; Samiul, S.H.; Kaiser, M.S.; Mahmud, M. Anomaly Detection in Electroencephalography Signal Using Deep Learning Model. In Proceedings of the International Conference on Trends in Computational and Cognitive Engineering, Dhaka, Bangladesh, 7–18 December 2020; pp. 205–217.
30. Domínguez-Morales, M.J.; Luna-Perejón, F.; Miró-Amarante, L.; Hernández-Velázquez, M.; Sevillano-Ramos, J.L. Smart footwear insole for recognition of foot pronation and supination using neural networks. *Appl. Sci.* **2019**, *9*, 3970. [CrossRef]
31. Luna-Perejón, F.; Muñoz-Saavedra, L.; Civit-Masot, J.; Civit, A.; Domínguez-Morales, M. AnkFall—Falls, falling risks and daily-life activities dataset with an ankle-placed accelerometer and training using recurrent neural networks. *Sensors* **2021**, *21*, 1889. [CrossRef] [PubMed]
32. Napoli, A.; Glass, S.M.; Tucker, C.; Obeid, I. The automated assessment of postural stability: Balance detection algorithm. *Ann. Biomed. Eng.* **2017**, *45*, 2784–2793. [CrossRef] [PubMed]
33. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Skeleton-based action recognition with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Kuala Lumpur, Malaysia, 3–6 November 2017; pp. 597–600.
34. Yoshikawa, Y.; Shishido, H.; Suita, M.; Kameda, Y.; Kitahara, I. Shot detection using skeleton position in badminton videos. In Proceedings of the International Workshop on Advanced Imaging Technology (IWAIT) 2021,International Society for Optics and Photonics, Online, 5–6 January 2021; Volume 11766, p. 117661K.
35. Nvidia. TRT_Pose Repository. Available online: https://github.com/NVIDIA-AI-IOT/trt_pose (accessed on 28 May 2021).
36. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *CoRR* **2014**, *8693*, 740–755.
37. Trt_pose_demo Repository. Available online: https://github.com/MACNICA-CLAVIS-NV/trt_pose_demo (accessed on 28 May 2021).
38. PyTorch Official Website. Available online: https://pytorch.org/ (accessed on 28 May 2021).

39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
40. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7291–7299.
41. Xiao, B.; Wu, H.; Wei, Y. Simple Baselines for Human Pose Estimation and Tracking. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 472–487.
42. Nvidia. Nvidia Tensor RT Official Website. Available online: https://developer.nvidia.com/tensorrt (accesed on 28 May 2021).
43. Nvidia. Official Repository of Tool *Torch2trt*. Available online: https://github.com/NVIDIA-AI-IOT/torch2trt (accesed on 28 May 2021).
44. Courbariaux, M.; Bengio, Y.; David, J.P. Training Deep Neural Networks with Low Precision Multiplications. *arXiv* **2015**, arXiv:cs.LG/1412.7024.