University of Cyprus
Department of Civil and Environmental Engineering

Master's Thesis

DETECTION OF HUMAN ACTION AND ERGONOMIC RISK ASSESSMENT AT CONSTRUCTION SITES BY USE OF MACHINE VISION AND DEEP LEARNING

Evagoras Lambrides

Nicosia, December 2022

Approval Form

Master's Thesis

DETECTION OF HUMAN ACTION AND ERGONOMIC RISK ASSESSMENT AT CONSTRUCTION SITES BY USE OF MACHINE VISION AND DEEP LEARNING

Presented by

Evagoras Lambrides

Supervisor: Symeon Christodoulou, Professor
Department of Civil and Environmental Engineering

____

Member of the Committee: Loukas J Dimitriou, Assistant Professor
Department of Civil and Environmental Engineering

____

Member of the Committee:

____

____

Copyrights

Acknowledgements

I would like to warmly thank my academic adviser Dr. Symeon Christodoulou for his undivided help as well as the moral support he provided me throughout my MSc thesis work.

I also want to thank my family, and especially my father, for their wholehearted support for the successful implementation of my thesis.

ABSTRACT

The construction industry is one of the most dangerous job sectors, in terms of health and safety. The dangerous construction-site environment and the unsafe behavior of construction workers inside this environment they operate in, often leads to serious injuries or even deaths. This was proved in many cases in the past. To reduce these incidences and to improve the safety performance of the construction workers, there is a need to identify and mitigate risk factors by continuously monitoring their behavior and by assessing the relative risks. The evolution of technology enables us to incorporate in the construction works new innovative methods which will greatly help in tackling the problem of health and safety.

This project focuses on the real-time detection and pose analysis of human activities at construction sites, as well as on the evaluation of the ergonomics of these activities. The pose detection and subsequent ergonomic analysis uses machine vision and deep learning technologies to detect human activities in images and/or video, and processes them by «skeletonizing» the detected worker pose, measuring the geometric properties of the pose's keypoints in the skeletal shape and calculating the corresponding scores according to the ***Rapid Entire Body Assessment (REBA) methodology***.

The proposed approach, which was successfully tested on several typical construction activities, has the potential of providing fast ergonomic assessment at construction sites. It also contributes to the knowledge of occupational safety and health in the construction industry, by providing a low-cost and accurate approach for assessing the risk factors of *Work-related Musculoskeletal Disorders (WMSDs).*

ΠΕΡΙΛΗΨΗ

Ο κατασκευαστικός κλάδος είναι ένας από τους πιο επικίνδυνους κλάδους εργασίας, όσον αφορά την υγεία και την ασφάλεια. Το εργασιακό περιβάλλον και η μη ασφαλής συμπεριφορά των εργαζομένων στα εργοτάξια είναι η γενεσιουργός αιτία πολλών ατυχημάτων και μπορεί να οδηγήσει σε σοβαρούς τραυματισμούς ή και θανάτους. Αυτό αποδείχθηκε σε πολλές περιπτώσεις στο παρελθόν. Για να μειωθούν αυτά τα περιστατικά και να βελτιωθούν οι επιδόσεις ασφάλειας των εργαζομένων στις κατασκευές, υπάρχει ανάγκη εντοπισμού και μετριασμού των παραγόντων κινδύνου με συνεχή παρακολούθηση της συμπεριφοράς τους και αξιολόγηση των σχετικών κινδύνων. Η εξέλιξη της τεχνολογίας μας δίνει τη δυνατότητα να ενσωματώσουμε στις κατασκευαστικές εργασίες νέες καινοτόμες μεθόδους που θα βοηθήσουν σημαντικά στην αντιμετώπιση του προβλήματος της υγείας και της ασφάλειας.

Η εργασία αυτή εστιάζει στον εντοπισμό και τη μελέτη των ανθρώπινων δραστηριοτήτων εντός των εργοταξίων σε πραγματικό χρόνο, καθώς και στην αξιολόγηση της εργονομίας αυτών των δραστηριοτήτων που εκτελούνται από τους εργαζόμενους. Αυτή η μέθοδος χρησιμοποιεί τεχνολογίες μηχανικής όρασης (machine vision) και τεχνητής νοημοσύνης (deep learning) για να ανιχνεύει ανθρώπινες δραστηριότητες μέσω βίντεο και τις επεξεργάζεται «σκελετοποιώντας» τους ανιχνευμένους εργαζόμενους, μετρώντας τις γεωμετρικές ιδιότητες των βασικών σημείων στο σχήμα του σκελετού και υπολογίζοντας τις αντίστοιχες βαθμολογίες σύμφωνα με τη μεθοδολογία ***Rapid Entire Body Assessment (REBA).***

Η προτεινόμενη προσέγγιση, η οποία δοκιμάστηκε με επιτυχία σε πολλές τυπικές κατασκευαστικές δραστηριότητες, έχει τη δυνατότητα να παρέχει γρήγορη εργονομική αξιολόγηση στα εργοτάξια. Συμβάλλει επίσης στη γνώση της επαγγελματικής ασφάλειας και υγείας στον κατασκευαστικό τομέα παρέχοντας μια χαμηλού κόστους και ακριβή προσέγγιση, για την αξιολόγηση των παραγόντων κινδύνου των *μυοσκελετικών διαταραχών που σχετίζονται με την εργασία (WMSDs).*

TABLE OF CONTENTS

ABSTRACT 5

ΠΕΡΙΛΗΨΗ 6

TABLE OF CONTENTS 7

LIST OF TABLES 8

LIST OF FIGURES 9

LIST OF ABBREVIATIONS 11

1. INTRODUCTION 12

2. RESEARCH BACKGROUND 18

2.1 Computer Vision (CV) 18

2.2 Work-related Musculoskeletal Disorders in Construction Industry 19

2.3 Rapid Entire Body Assessment (REBA) 21

3. RESEARCH METHODOLOGY 26

3.1 Pose Estimation 27

3.2 REBA 29

4. RESULTS 31

5. SUMMARY OF FINDINGS 61

6. CONCLUSIONS 63

REFERENCES 64

APPENDIX 65

LIST OF TABLES

Table 2.1: REBA scores 25

Table 3.1: Coordinates of REBA joints of interest, AFTER re-referencing 28

Table 3.2: Human pose analysis flowchart 30

Table 4.1: Pose 1, software code results 33

Table 4.2: Pose 2 Code results 34

Table 4.3: Pose 3 Code results 37

Table 4.4: Pose 4 Code results 40

Table 4.5: Pose 5 Code results 43

Table 4.6: Pose 6 Code results 46

Table 4.7: Pose 7 Code results 49

Table 4.8: Pose 8 Code results 52

Table 4.9: Pose 9 Code results 55

Table 4.10: Pose 10 Code results 58

LIST OF FIGURES

Figure 1.1: Construction compared to industries with similar work activities 12

Figure 2.1: Action Recognition using Computer Vision 19

Figure 2.2: WMSDs by body parts 20

Figure 2.3: REBA employee Assessment Worksheet.. 23

Figure 3.1: Before and after analysis human pose 26

Figure 3.2: Mediapipe analysis body parts 27

Figure 4.1: Before and after pose-estimation analysis; (a) Worker pose 1, (b) Skeletonized pose showing body part keypoints. 31

Figure 4.2: Pose 1, manual evaluation results.. 32

Figure 4.3: Before and after pose-estimation analysis; (a) Worker pose 2, (b) Skeletonized pose showing body part keypoints. 34

Figure 4.4: Pose 2 Manual results.. 35

Figure 4.5: Before and after pose-estimation analysis: (a) Worker pose 3, (b) Skeletonized pose showing body part keypoints. 37

Figure 4.6: Pose 3 Manual results.. 38

Figure 4.7: Before and after pose-estimation analysis: (a) Worker pose 4, (b) Skeletonized pose showing body part keypoints. 40

Figure 4.8: Pose 4 Manual results.. 41

Figure 4.9: Before and after pose-estimation analysis: (a) Worker pose 5, (b) Skeletonized pose showing body part keypoints. 43

Figure 4.10: Pose 5 Manual results.. 44

Figure 4.11: Before and after pose-estimation analysis: (a) Worker pose 6, (b) Skeletonized pose showing body part keypoints. 46

Figure 4.12: Pose 6 Manual results..47

Figure 4.13: Before and after pose-estimation analysis: (a) Worker pose 7, (b) Skeletonized pose showing body part keypoints. 49

Figure 4.14: Pose 7 Manual results..50

Figure 4.15: Before and after pose-estimation analysis: (a) Worker pose 8, (b) Skeletonized pose showing body part keypoints. 52

Figure 4.16: Pose 8 Manual results..53

Figure 4.17: Before and after pose-estimation analysis: (a) Worker pose 9, (b) Skeletonized pose showing body part keypoints. 55

Figure 4.18: Pose 9 Manual results..56

Figure 4.19: Before and after pose-estimation analysis: (a) Worker pose 10, (b) Skeletonized pose showing body part keypoints. 58

Figure 4.20: Pose 10 Manual results..59

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CV	Computer Vision
DL	Deep Learning
REBA	Rapid Entire Body Assessment
WMSDs	Work-related Musculoskeletal Disorders

1. INTRODUCTION

Health and Safety has always been one of the biggest problems in the construction sector causing a large number of accidents around the world. The most common cause of this problem is the unsafe way construction workers operate and the non-observance of the necessary protection measures in construction sites, a combination that can be catastrophic. Uncomfortable work postures, repetitive and heavy lifting, and excessive force or overexertion are some ergonomic risk factors that can lead workers to develop work-related musculoskeletal disorders (WMSDs). In 2021 there were approximately 40.000 cases of work-related musculoskeletal disorders in the United Kingdom (according to the Health and Safety Executive, HSE). Also, according to the records of the International Labor Organization (ILO), every year about 318.000 work-related accidents occur, with a substantial part of them being related to the construction sector.

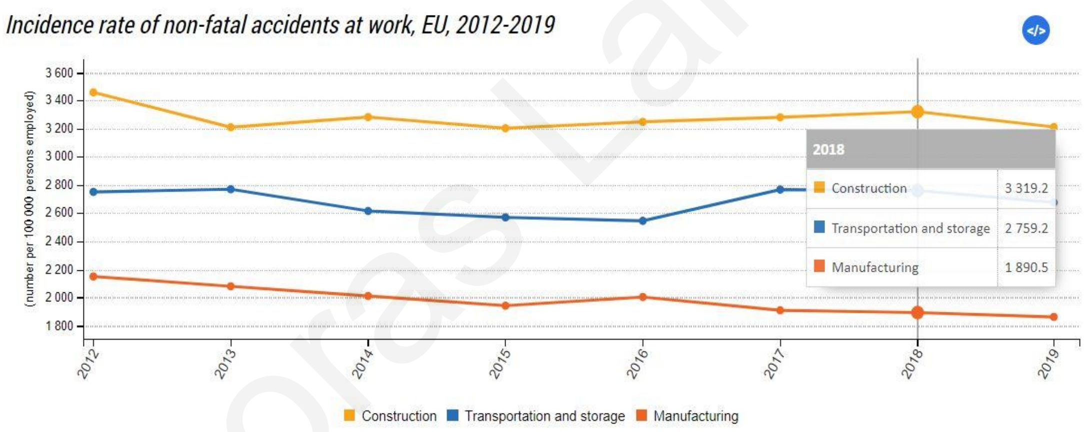


Figure 1.1: Construction compared to industries with similar work activities

The issue of safety at the workplace has always concerned the workers and especially workers in the construction sector, where most accidents are usually observed. One of the most recent and important examples of construction site accidents is the preparation for the World Cup in Qatar, where according to another ILO study 50 migrant workers died, 500 migrants were seriously injured and 37.600 suffered mild to moderate injuries in 2020. The main causes of serious injuries were falls, car accidents and falling objects. Incidence rate of non-fatal accidents at work, in European Union during the period from 2012 to 2019 is shown in the figure above [1].

Considering all the above, we understand the need for the inclusion of an improved and more effective method of monitoring construction work, which will aim to obtain a more complete picture of work behaviour, so that the necessary steps can be taken to minimize accidents. The classic manual (and intermittent) inspections using construction site foremen have proven not only ineffective in terms of time and cost but also less accurate, making the integration of technology in construction industry necessary.

Over time, several investigations have been carried out for identifying and understanding the causes of accidents, as well as the methods of dealing with this problem.

In the work of *S. Hignett and L. McAtamney (1999), "Rapid Entire Body Assessment (REBA)"*, the authors worked entirely with the Rapid Entire Body Assessment (REBA) postural analysis tool, and how to apply it through examples. REBA is reported to have been developed as a field tool specifically designed to identify the various types of unpredictable working postures encountered in health care and other service industries. Data were collected from over 600 postural examples to establish the body part ranges in the REBA score sheets. After that, the sensitizing concepts of load, coupling, and activity were then incorporated to produce the final REBA score (1–15), with accompanying levels of risk and action levels. The authors conclude that although the initial development of REBA shows promises as a useful postural analysis tool, further validation needs to be carried out. It is also mentioned that others may be better placed to carry out this validation, perhaps in cross reference to other tools (OWAS, NIOSH, posture targeting, biomechanical models) or through empirical measurement in a laboratory setting.

In *M. Massiris Fernandez et al.'s "Ergonomic risk assessment based on computer vision and machine learning" (2020)*, the authors dealt with a topic similar to the work discussed in this thesis, presenting a method that performs accurate ergonomic risk assessment and that automatically computes Rapid Upper Limb Assessment (RULA) scores from snapshots or digital video using computer vision and machine learning techniques. It is reported that this method can also handle multiple workers simultaneously, even under sub-optimal viewing conditions. In this case, the RULA tool is used instead of the REBA tool that is the focus of this thesis's research work. The processing workflow uses open-source neural networks to detect the workers' skeletons, after which their body-joint positions and angles are inferred, with which RULA scores are computed. As reported, the method was validated in actual outdoor working situations under the technical supervision of seven experienced ergonomists, who also evaluated the associated RULA scores. The validation methods involved three levels of comparison:

1. Skeleton and joint detection confidences by viewpoint

2. Angle comparison between lab-controlled and simulated viewpoints,

and

3. RULA score agreement across the proposed method and observations from experienced ergonomists.

The author concludes that the experimental results, provide positive evidence regarding the feasibility of the method, and it is also mentioned that reasonable variations in camera view do not influence the results in real working conditions significantly. Finally, according to the paper there are two potential weaknesses that may lead to errors. Skeleton detection biases in some cases may lead to relevant angle measurement deviations and also, the angular measurements are not computed from 3D body-joint estimates, but from 2D projections, which may raise projective distortions.

In *H.Guo et al.'s "Image-and-Skeleton-Based Parameterized Approach to Real-Time Identification of Construction Workers' Unsafe Behaviours"* (2018), the authors talk about the unsafe behaviours of site workers and about what can be done to prevent construction accidents. The authors present a skeleton-based real-time identification method by combining image-based technologies, construction safety knowledge and ergonomic theory. The proposed method recognizes unsafe behaviours by simplifying dynamic motion into static postures, which can be described by a few parameters. Three basic modules are involved: the unsafe behaviour database, real-time data collection module and behaviour judgement module.

A laboratory test demonstrated the feasibility, efficiency and accuracy of the method. The test described in the paper, is about climbing the ladder of concrete mixer truck which is said to be one of the most common unsafe behaviours according to construction accident statistics. After that, six parameters were chosen to describe the every posture on the ladder, including 1) the angle of the left elbow, 2) angle of the right elbow, 3) angle of the left knee, 4) angle of the right knee, 5) inclination angle of the upper body and 6) the inclination angle of the lower body.

According to the authors this experiment demonstrates that the method is feasible, accurate and efficient and also mention four main advantages:

1) Real-time identification
2) Invariance of view
3) Non-intrusiveness
4) Intuitive spatial features

However, the authors state that the method still needs to be improved in some aspects such as the following:

1) A complete unsafe behaviour sample database has not yet been established.
2) The test only considered the motion of one worker.
3) More tests and on-site experiments are needed to ensure the reliability of the parameter value ranges.
4) The proposed method requires image capturing in real time on construction sites.

In the work by *M. Fordjour Antwi-Afari et al. (2018), "Wearable insole pressure system for automated detection and classification of awkward working postures in construction workers"*, the authors refer, as in the previously noted papers, to the awkward working postures as the main risk factor for work-related musculoskeletal disorders (WMSDs) in construction. Their study developed a method to automatically detect and classify awkward working postures based on foot plantar pressure distribution data measured by a wearable insole pressure system. In order to apply the method ten asymptomatic participants performed five different types of awkward working postures (overhead working, squatting, stooping, semi-squatting, and one-legged kneeling) in a laboratory setting. Four supervised machine learning classifiers (artificial neural network (ANN), decision tree (DT), K-nearest neighbor (KNN), and support vector machine (SVM)) were compared and the best was used for classification performance using a 0.32s window size.

After the experiment, results showed that the SVM classifier obtained the best results with 99.70% accuracy, and the sensitivity of correctly classifying each awkward working posture was above 99.00%. It was further noted that the method has the potential to allow safety managers to continuously monitor and minimize workers' exposure to awkward working postures on construction sites.

Finally the authors point to some limitations of the method. First, experiments were designed and conducted to only include simulated awkward working postures in a homogenous sample. Other risk factors should be examined in the future.

It is also unknown whether other biomechanical exposures such as repetitive motions, high force exertions and vibration will affect foot plantar pressure distribution data captured by a wearable insole pressure system.

In the paper *"Sensing construction work-related musculoskeletal disorders (WMSDs)." by Alwasel et al. (2011)*, the authors deal with the overall problem of work-related musculoskeletal disorders (WMSDs) in construction emphasizing on what affects a worker's shoulders. It also presents a background on the kinematics of shoulder movement and explains the biomechanics and the causes of shoulder injuries. Then, the authors present preliminary results for a prototype of a simple, low-cost, sensing solution for automatically monitoring undesirable movements and patterns of motion, which is expected to reduce Construction WMSDs. The proposed method requires the use of a magnetoresistive angle sensor to measure human joint angles. According to the author this is a unique solution for this problem combining accurate measurements, low-cost, and applicability to wide-scale field deployment.

The sensor system can be used to monitor workers' onsite exposure to dangerous postures providing data that can be used to help reduce WMSDs. The measurements can also inform efforts to redesign a workplace to make it ergonomically safer for workers.

As aforementioned, the research work discussed in this thesis focuses on the study and evaluation of human poses during the execution of work at construction sites (using the Rapid Entire Body Assessment, REBA), and on the assessment of Work-related Musculoskeletal Disorders (WMSDs).

To this effect, I present an automated method of monitoring and evaluating the way workers work on site to determine if staff are working safely, and if there is a risk of injury. The method is based on Computer Vision (CV) and Deep Learning (DL) technologies and uses software code (in Python) to detect and analyse human activity in videos or photos, of workers at a construction site. The data is processed by "skeletonizing" the detected workers, measuring the geometric properties of keypoints in the skeletal shape and by calculating the corresponding scores according to REBA. The near-real-time analysis of a worker's pose and the obtained metrics and statistics can help us draw information and conclusions about the degree of safety at which workers operate, so that appropriate measures can be taken to reduce potential health risks to them.

The thesis contains a "*Research Background*" chapter, partially discussing the thesis's three main subjects (CV, WMSDs, REBA) as a backdrop to the subject work. The chapter is followed by a chapter on "*Research Methodology*", in which human pose estimation and the application of REBA are discussed. Detailed results follow, which include the software code's output, the manual calculations (for comparison) and an explanation for each case studied. The final two chapters are the "*Summary of Findings*" and "*Conclusion*", in which the work's findings are summarized and discussed.

2. RESEARCH BACKGROUND

2.1 Computer Vision (CV)

CV is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. In other words, this technology enables computers to "see" the world then make decisions or gain understanding about the environment and situation. [2]

CV can be compared to human vision in terms of how it works, but with the difference that in humans there is the advantage of time. Human vision has the time to process and understand what it sees in order to finally draw the appropriate conclusions, unlike the CV which you are called upon to receive and analyse thousands of data using algorithms in a much shorter time. This is its main advantage since in this way it exceeds human capabilities. [3]

Computer vision can have many applications in the field of construction others than health and safety such as the following: [4]

- Building Information Modelling (BIM)
- Keep project costs under control
- Monitoring productivity
- Construction safety management
- Addressing staff and manpower shortage
- Green building construction
- Predictive maintenance

The following figure shows an example of human action recognition using CV [5].

Figure 2.1: Action Recognition using Computer Vision

2.2 Work-related Musculoskeletal Disorders in Construction Industry

Work-related Musculoskeletal Disorders (WMSDs) is a group of painful disorders of muscles, tendons, and nerves that are caused, or aggravated by, various difficult work tasks. Carpal tunnel syndrome, tendonitis, thoracic outlet syndrome, and tension neck syndrome are some examples. The construction industry is one of the leading sectors in which WMSDs are strongly observed, with thousands of workers worldwide showing symptoms of the disorder. Workers such as rebar workers, bricklayers and roofers are, by virtue of their occupation, frequently exposed to elevated physical risk factors. The most important WMSD risk factors are related to lifting weights and to awkward postures because such actions require maintaining muscle force over an extended period of time. Repetitive and prolonged lifting tasks cause muscle fatigue and discomfort for a worker and invariably this activity increases the risk of developing WMSDs [6]. In the United States alone, there were 79,890 cases of musculoskeletal disorders in construction work between 2013 and 2015, while in the UK during the same period there were 990 cases were of back disorders from 100,000 construction workers [7].

Notably WMSDs not only lead to worker ill-health but also to reduced productivity and concomitant financial loss. Therefore, risk factors associated with WMSDs should be identified in order to develop effective ergonomic interventions to prevent WMSDs in construction workers. [6]

Unfortunately, at many construction sites, the control by the project managers of the workers' actions and safety is usually limited to the avoidance of fatal accidents and not necessarily specific injuries or of long-term effects of them. This obviously does not help in the treatment of WMSDs, and by extension in the adoption and integration of new methods of work monitoring at the construction sites.

It is also worth noting that some health and safety organizations, such as the Occupational Safety and Health Administration (OSHA) and the National Institute for Occupational Safety and Health (NIOSH), have promoted general ergonomic best-practices to reduce the risk of WMSDs occurring among workers.

The following graph shows how the different body parts get affected from WMSDs by percentage [7]:

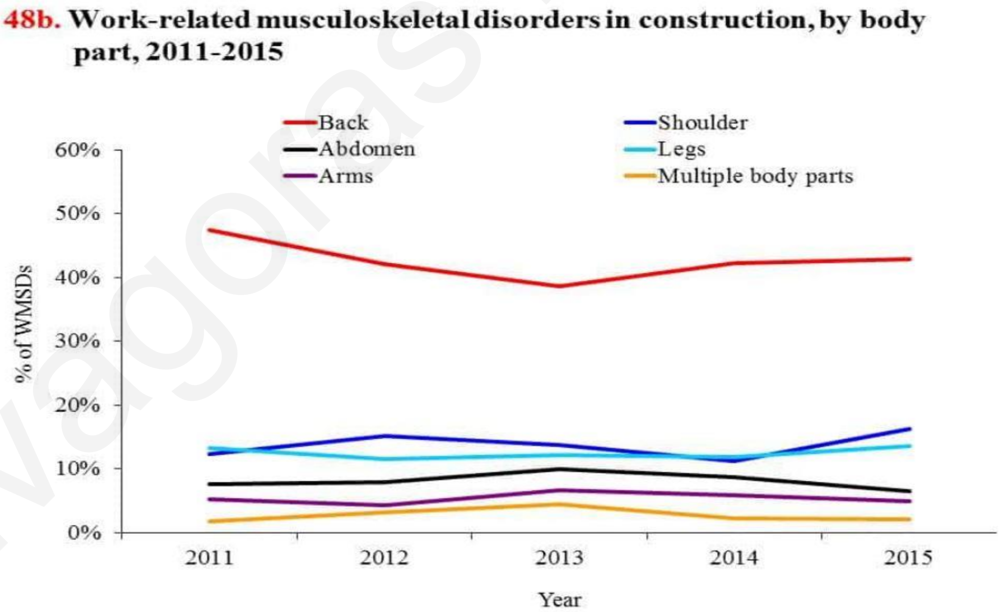


Figure 2.2: WMSDs by body parts

2.3 Rapid Entire Body Assessment (REBA)

To implement a new, automated, method of work supervision in order to prevent accidents and injuries, we need an appropriate way of evaluating the behaviour of workers based on their movements and postures.

The Rapid Entire Body Assessment (REBA) is a tool used to evaluate the risk of musculoskeletal disorders (MSDs) associated with specific tasks within a job. It is a whole-body screening tool that follows a systematic procedure to assess biomechanical and postural loading on the body. The benefits of this tool are that it is simple, quick, and requires minimal equipment (pen and paper) making it easy to complete multiple assessments per task or per job. REBA evaluates the whole body, and it can be used to assess any task [8]. In this research project, REBA will be applied using coding (in the Python programming language) instead of the traditional way (pen and paper), so that the digital data to be received can be utilized in an automated manner.

REBA was created based on the following objectives: [9]

1. Provide a simple postural analysis system sensitive to musculoskeletal risks in a variety of tasks.

2. To divide the body into segments to evaluate individually with reference to postures and movement planes.

3. Provide a scoring system for muscle activity caused by static, dynamic, rapid changing or unstable postures.

4. To consider coupling as an important variable in the handling of loads.

5. To give an action level output with an indication of urgency.

6. To provide a user-friendly assessment tool that requires minimal time, effort, and equipment.

How to use REBA:

Usually, for the preparation of a REBA assessment the evaluator does a research to understand the tasks and work requirements of the person to be evaluated, observing his movements and postures during several work cycles. Selection of postures to be evaluated should be based on (1) the most difficult postures and work tasks (based on the employee's interview and initial observation), (2) the posture maintained for the longest period or (3) the posture where the higher the force loads occur.

By using REBA in a computerized way, instead of the traditional way (pen and paper), multiple positions and tasks within the work cycle can be evaluated in a very short time and with negligible effort by the evaluator. [9]

To complete the REBA Assessment Worksheet (Figure 2.3 [9]), the assessor first evaluates the 'Group A' postures regarding the worker's trunk, neck and legs. Then the postures of 'Group B' for the upper and lower arm and the wrist are evaluated. For each body region, there is a posture-scoring scale plus adjustment notes for additional considerations. The assessor then rates the 'load / strength' and 'coupling factors'.

Finally, the person's activity is evaluated. The scores for 'Group A' and 'Group B' are found in Tables A and B respectively. Score A is the sum of the 'Table A' score and the 'Load / Force' score. Score B is the sum of the 'Table B' score and the 'Coupling' score for each hand. Score C is read from 'Table C', by entering it with 'Score A' and 'Score B' [10].

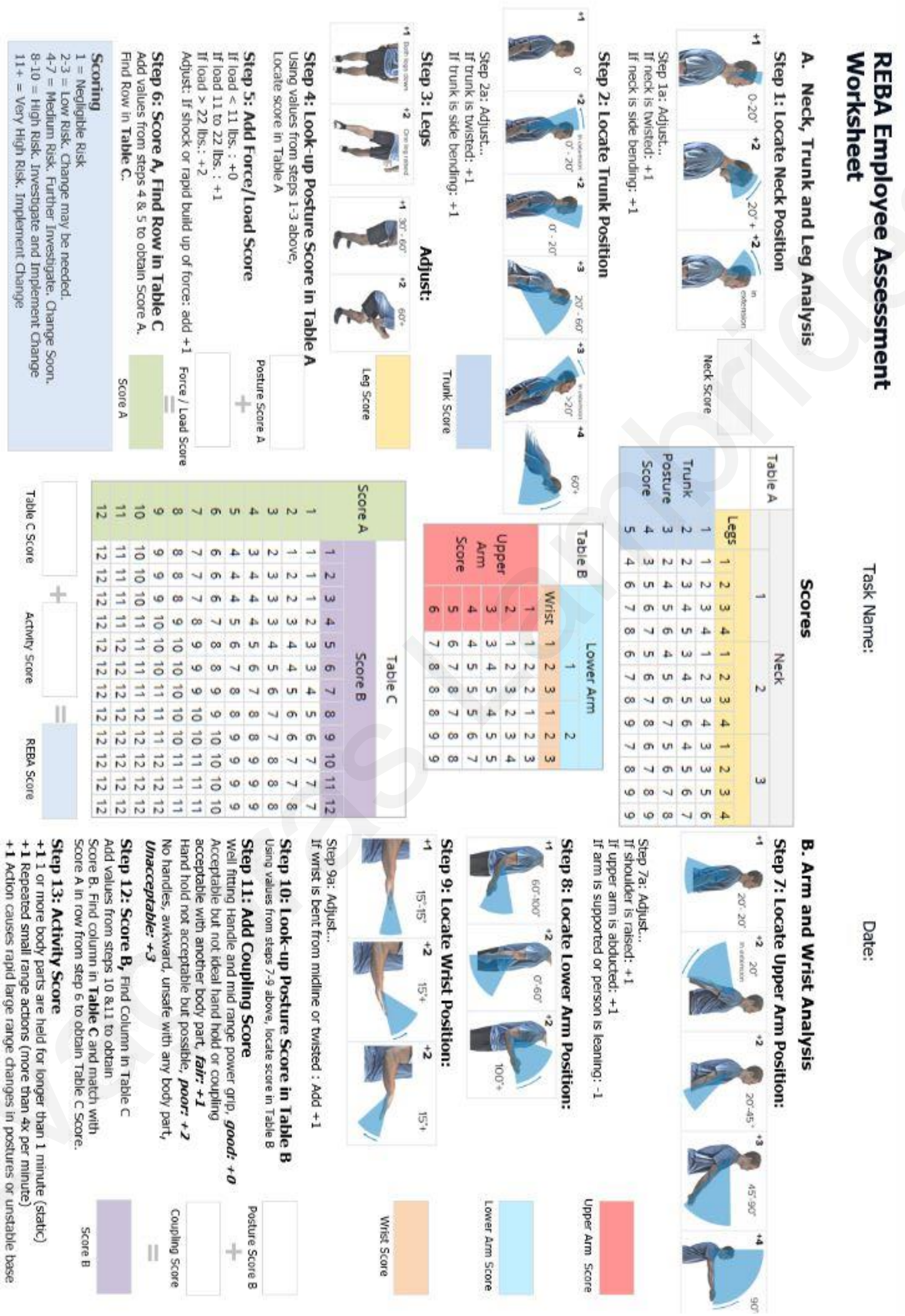
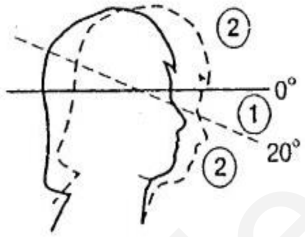


Figure 2.3: REBA employee Assessment Worksheet

- Group A body part diagrams [11]:

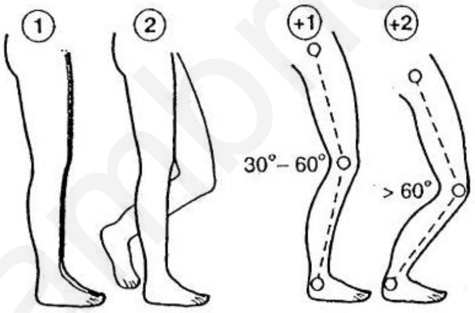
Neck

Movement	Score	Change score:
0°–20° flexion	1	+1 if twisting or side flexed
>20° flexion or in extension	2	



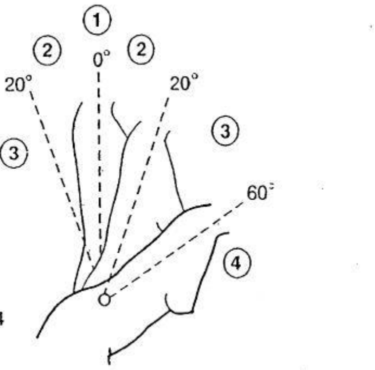
Legs

Position	Score	Change score:
Bilateral weight bearing, walking or sitting	1	+1 if knee(s) between 30° and 60° flexion
Unilateral weight bearing Feather weight bearing or an unstable posture	2	+2 if knee(s) are >60° flexion (n.b. Not for sitting)

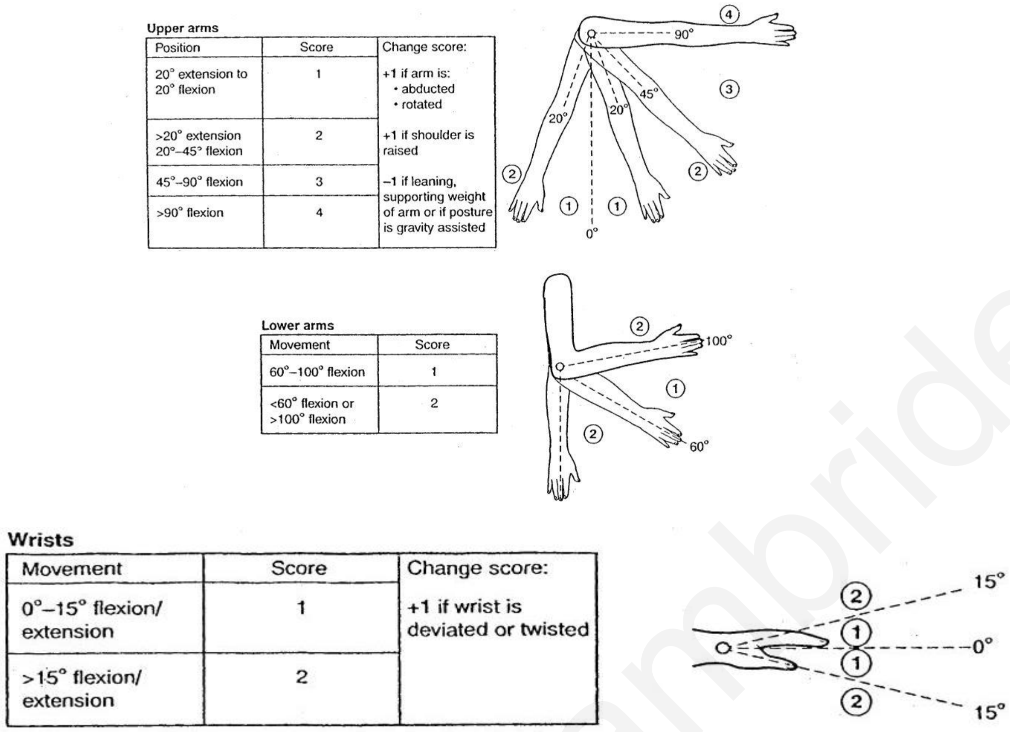


Trunk

Movement	Score	Change score:
Upright	1	+1 if twisting or side flexed
0°–20° flexion 0°–20° extension	2	
20°–60° flexion >20° extension	3	
>60° flexion	4	



➢ Group B body part diagrams [11]:



The application of REBA gives us a final single REBA score (in the range of 1 to 15), calculated as the sum of 'Score C' and the 'Activity' score and which represents the level of MSD risk for the work being evaluated. The below table describes the scores in more detail [9].

Table 2.1: REBA scores

SCORE	Level of MSD Risk
1	Negligible risk, no action required
2-3	Low risk, change may be needed
4-7	Medium risk, further investigation, change soon
8-10	High risk, investigate and implement change
11+	Very high risk, implement change

3. RESEARCH METHODOLOGY

As mentioned in the introduction, the methodology on which this research work is based on mainly includes the use of a programming language (Python) and the application of several computer vision (**CV)** and deep-learning **(DL)** technologies. The objective was to create an automated software that would receive sensory data (images), process them to recognize all body parts of the imaged (worker) and finally to apply the REBA approach to compute the rating of, and thus the hazard in, the body posture in investigation.

The detection and identification of the human body through the input (images or videos) presented to the software program, was the most difficult part of the study since for the correct operation of the program the input image needs to be processed in three dimensions (both x,y,z coordinates and joint angles are required). From the analysis of each input image (such as the one shown in Figure 3.1a), a skeletonized pose is deduced (Figure 3.1b) and coordinates (in 3 dimensions) of keypoints are extracted, representing the various body parts and body joints.

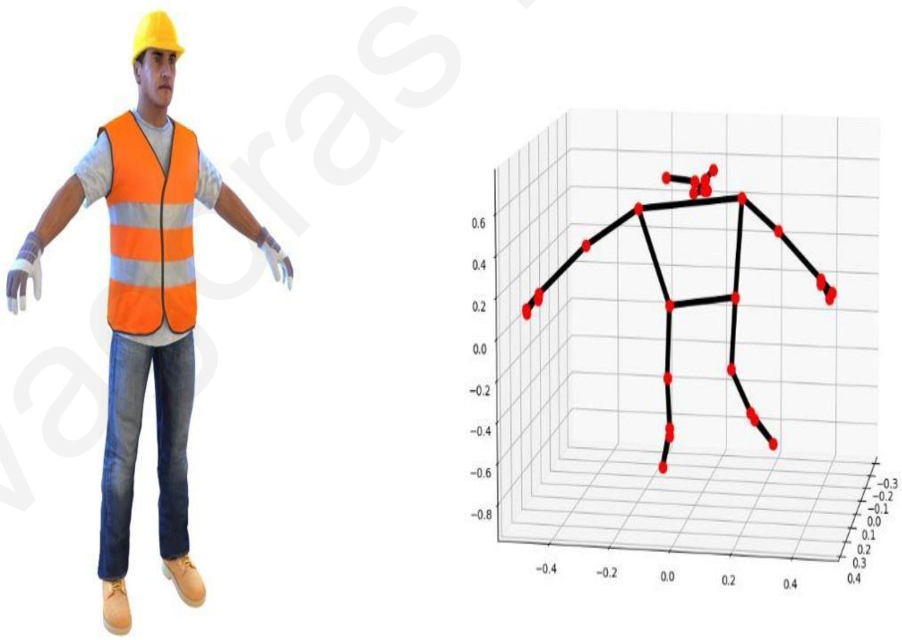


Figure 3.1: Before and after analysis human pose

The developed software code (Appendix) is based on readily available code modules and is composed of two key steps: firstly, an input image is processed, the human pose in it is extracted in 3D and the coordinates of the various body parts are estimated (part 1); then, application of REBA (part 2) is performed based on the coordinates extracted from the previous step, the sub and total scores are computed, and then the final evaluation is deduced.

3.1 Pose Estimation

Pose estimation is performed by use of the *OpenCV* and *Mediapipe* machine vision libraries. *OpenCV* is a general-purpose machine-vision library and *Mediapipe* is a framework for building machine-learning pipelines for processing time-series data such as video and audio, which offers ready-to-use yet customizable Python solutions as a prebuilt Python package. The aforementioned Python libraries allow the computation and extraction of 3D coordinates (x,y,z) for 33 different points on a human body (shown below in Figure 3.2) which in turn allow for the deduction of the human pose [12].

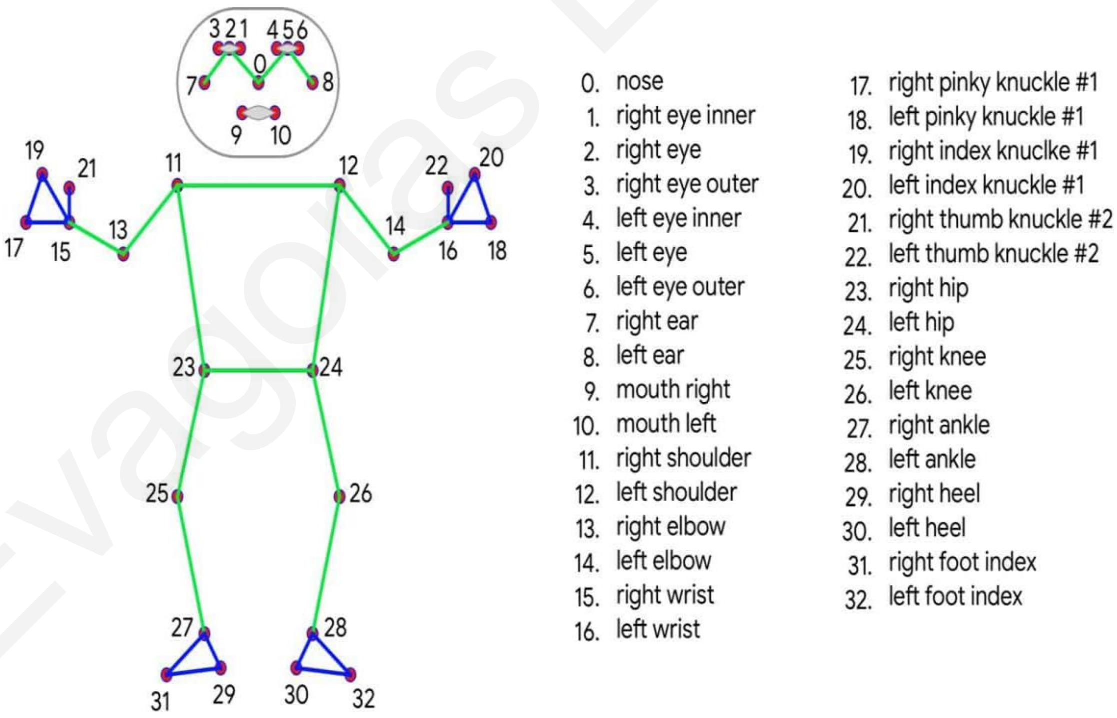


Figure 3.2: Mediapipe analysis body parts

The application of Mediapipe in Python is done by first importing the appropriate package ("*import Mediapipe*") into the developed software code, which gives the user access to all the human-pose estimation libraries needed for the computations.

It should be noted, though, that the selection of *Mediapipe* for performing the pose estimation presents a problem with regard to the number of body parts it computes. More specifically, using *Mediapipe*, as mentioned above, 33 body parts are extracted, while REBA uses 16 of them to apply its analysis and to extract the results. Thus, a down-sampling and/or remapping of the deduced body parts is required. The problem was solved by creating a mapping between the 33 body parts from *Mediapipe* to the 16 body parts used by REBA, so that we get as a result the coordinates of only the 16 keypoints needed. After the pose estimation process is completed, for each different case of a figure (pose) that we analyse in the code, we get as a result the 3D coordinates (x,y,z) for each of the 16 body parts that are needed, and which constitute a complete human figure. At the outset of this stage we are able to apply the REBA code, now having as input the coordinates of the body parts instead of an image.

Table 3.1: Coordinates of REBA joints of interest, AFTER re-referencing

Coordinates of REBA joints of interest, AFTER re-referencing w.r.t. hips			
REBA		**MediaPipe**	
0:	Head	1:	left_eye_inner
1:	Nose	0:	Nose
2:	LShoulder	11:	left_shoulder
3:	LElbow	13:	left_elbow
4:	LWrist	15:	left_wrist
14:	LHand (optional)	19:	left_index
5:	RShoulder	12:	right_shoulder
6:	RElbow	14:	right_elbow
7:	RWrist	16:	right_wrist
15:	RHand(optional)	20:	right_index
8:	LHip	23:	left_hip
9:	LKnee	25:	left_knee
10:	LAnkle	27:	left_ankle
11:	RHip	24:	right_hip
12:	RKnee	26:	right_knee
13:	RAnkle	28:	right_ankle

3.2 REBA

Before the REBA code is applied, the coordinates we received from Mediapipe which are in .txt format should be converted into .json files since in this case the code input is required to be in this format. This process was easily performed using windows command prompt and the "*rename <name.txt> <name.json>*" command.

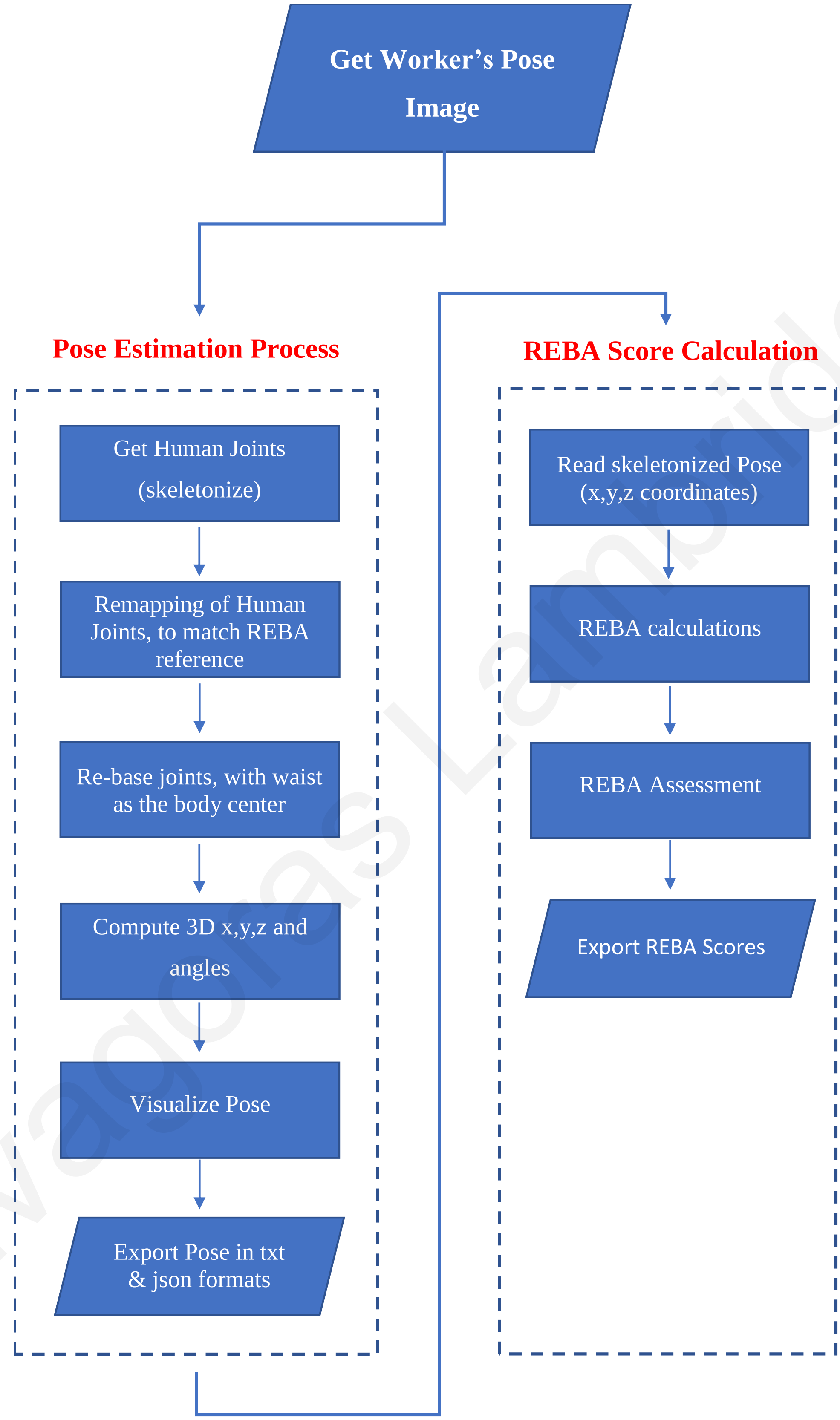
From this analysis of the code, we get as a result the partial and overall scores of Part A (Neck, Trunk and Leg analysis) and Part B (Arm and Wrist Analysis) of the table, and then the score of table C which in combination with the Activity Score leads us to the final REBA Score.

The overall Score A and Score B are derived after combining with the scores for Force/Load and Coupling respectively. However, the code's calculations are entirely based on the posture of the body being studied (body parts coordinates) and therefore it does not have the ability to calculate Force/Load and Coupling scores. For this reason, these 2 scores are considered equal to 0, which does not affect the correctness of the results since the main purpose of the project is the study of body posture and whether it will affect the health of the workers. The same goes for the Activity Score. Moreover, the code separates its ratings, by showing in result only the highest score for each body part. For instance, if in a certain pose the left upper arm gets a higher score than the right upper arm, the code will take as a result the left upper arm score. This, makes sure that only the most "dangerous" movements in the worker's pose are evaluated.

Both Python Codes (Pose Estimation [13], REBA calculations [14]) where received in their original version from GitHub, and after a few modifications (e.g. coordinates re – referencing) where used for the project. GitHub, is an internet hosting service software for distributed version control which uses a free and open source software called Git. It is commonly used to host open source software development projects.

The following flowchart shows a quick summary of the entire process to analyze a pose using the code and get the final REBA scores.

Table 3.2: Human pose analysis flowchart



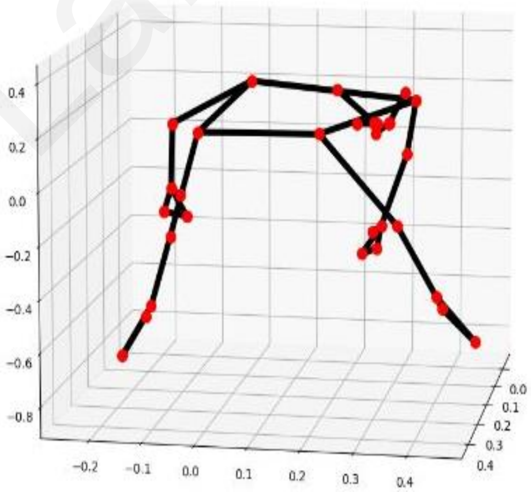
4. RESULTS

The method used in this project was applied in the way described above, and case-studied on different poses of workers at the construction site. These poses were studied, using REBA, in order to evaluate their dangerousness in terms of worker's health.

In each different case considered, the calculations for the final REBA score and by extension the pose's risk to a worker's health, were performed by using both the code (Python) and the traditional way (by filling in the REBA table by hand). This way, one can compare the two methods with each other (software code, manual method), their similarities and differences, their suitability to task and their reliability.

POSE 1:

(a)



(b)

Figure 4.1: Before and after pose-estimation analysis; (a) Worker pose 1, (b) Skeletonized pose showing body part keypoints.

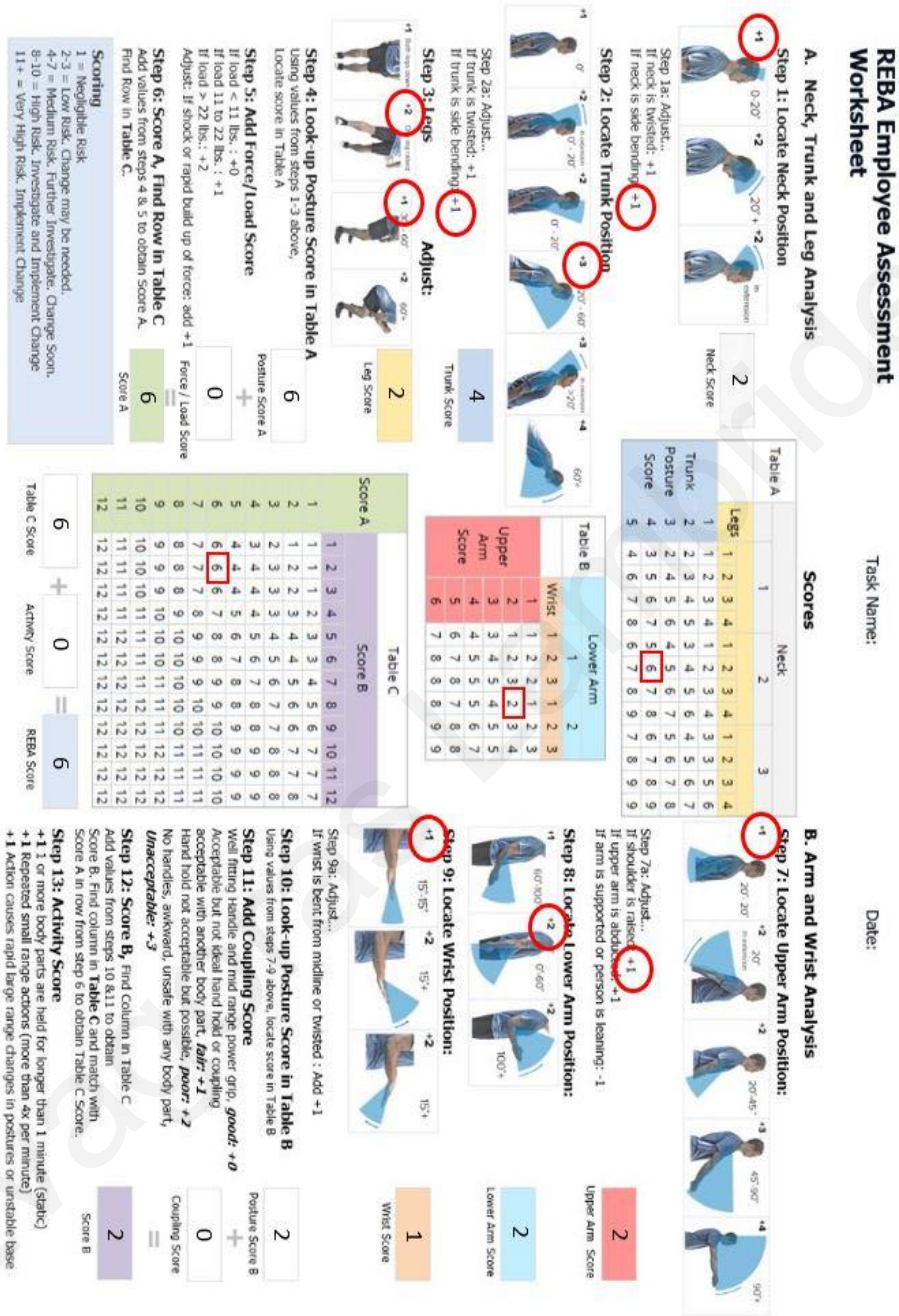


Figure 4.2: Pose 1, manual evaluation results

Code Results:

Table 4.1: Pose 1, software code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">5</td><td colspan="3">4</td><td>5</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">Medium Risk. Further Investigate. Change Soon</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>2</td><td>2</td><td>3</td></tr>
</table>

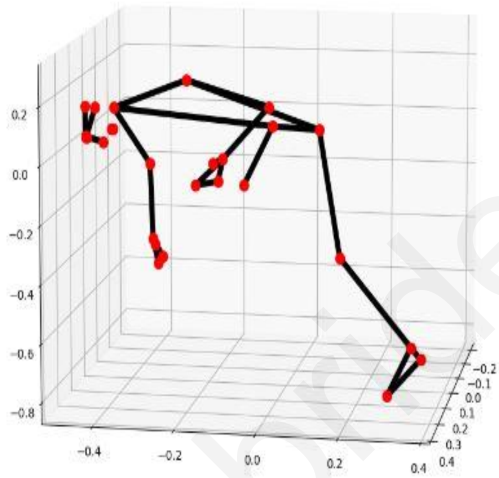
Discussion:

Starting the analysis with the first pose, we notice that the software code's results are very close to those of the manual analysis with a final REBA score of '5' and '6' respectively. In step 1, both methods produce a neck score of '2', since the head inclination varies between 0-20 degrees towards the front and the neck is twisted, while in step 2 (Trunk position) the manual approach gives a score of '4' and the code a score of '3'. The difference here is likely to be the '+1' mark for the trunk side bending, which was taken into consideration in the manual calculation. Regarding the leg position, we have bilateral weight bearing and an inclination between 30-60 degrees, so the leg score equals to '2'.

Observing the hands, we can see inclination angles in the range of -20 to +20 degrees for the upper arm and 0 to 60 degrees for the lower arm. The wrist position shows a considerable deviation in the score given by the code (1 pt in the manual calculation and 3pt in the code). In the photo, the wrist does not seem to exceed the limit of -15 + 15 degrees, so in this case the manual rating is more correct.

POSE 2:

(a)



(b)

Figure 4.3: Before and after pose-estimation analysis; (a) Worker pose 2, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.2: Pose 2 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">8</td><td colspan="3">5</td><td>10</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">High Risk. Investigate and Implement Change</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>3</td><td>3</td><td>4</td><td>3</td><td>2</td><td>3</td></tr>
</table>

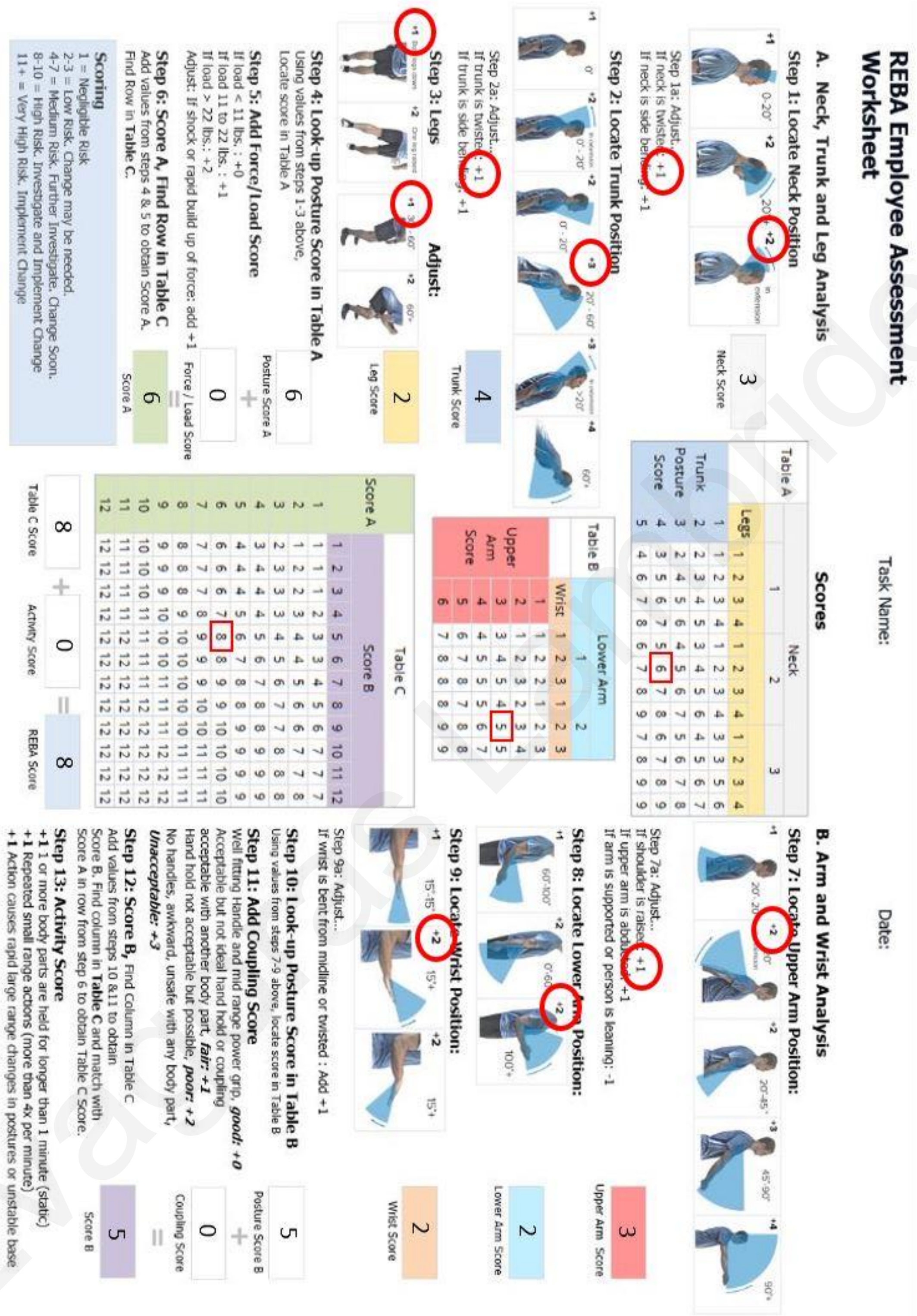
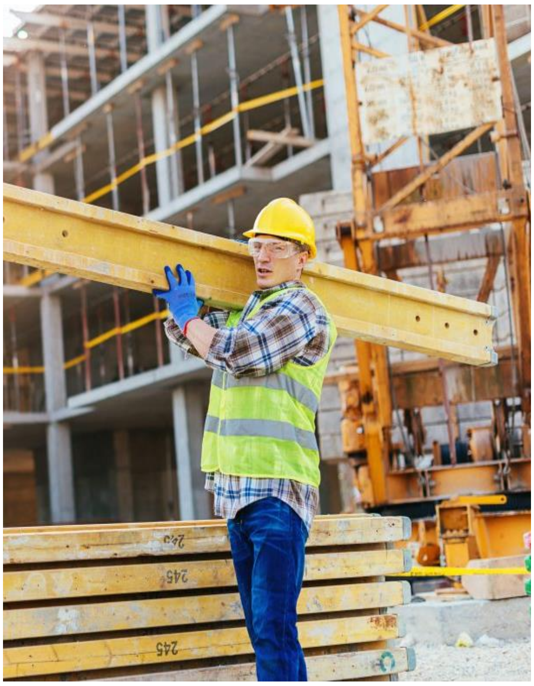


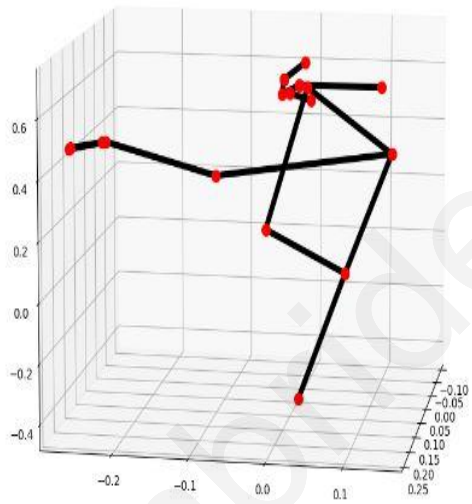
Figure 4.4: Pose 2 Manual results

Discussion:

In pose 2, we have a 2-point deviation in the scores, with a score of ‘10’ in the code and of ‘8’ in the manual solution. The biggest difference in this case is spotted in the leg score with a manual score of ‘2’ and code score of ‘4’. The neck is turned backwards and also twisted (+3), while the trunk position is estimated at 20-60 degrees and twisted (+4) with a difference of one point from the code score. With regard to the arms, we have an upper arm in a negative inclination with the elbow raised (+3) and the lower arm in a position that exceeds 100 degrees (+2). The wrist is estimated to be in a position of 15+ degrees up (+2) and there is a 1 point difference from the code which scores it with +3. The extra point probably is due to the possibility that the wrist is bent from midline or twisted.

POSE 3:





(a) (b)

Figure 4.5: Before and after pose-estimation analysis: (a) Worker pose 3, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.3: Pose 3 Code results

SCORE A			SCORE B			SCORE C
9			**7**			**11**
Partial			Partial			Very High Risk. Implement Change
Neck	Trunk	Legs	Upper Arm	Lower Arm	Wrist	
3	5	4	4	2	3	

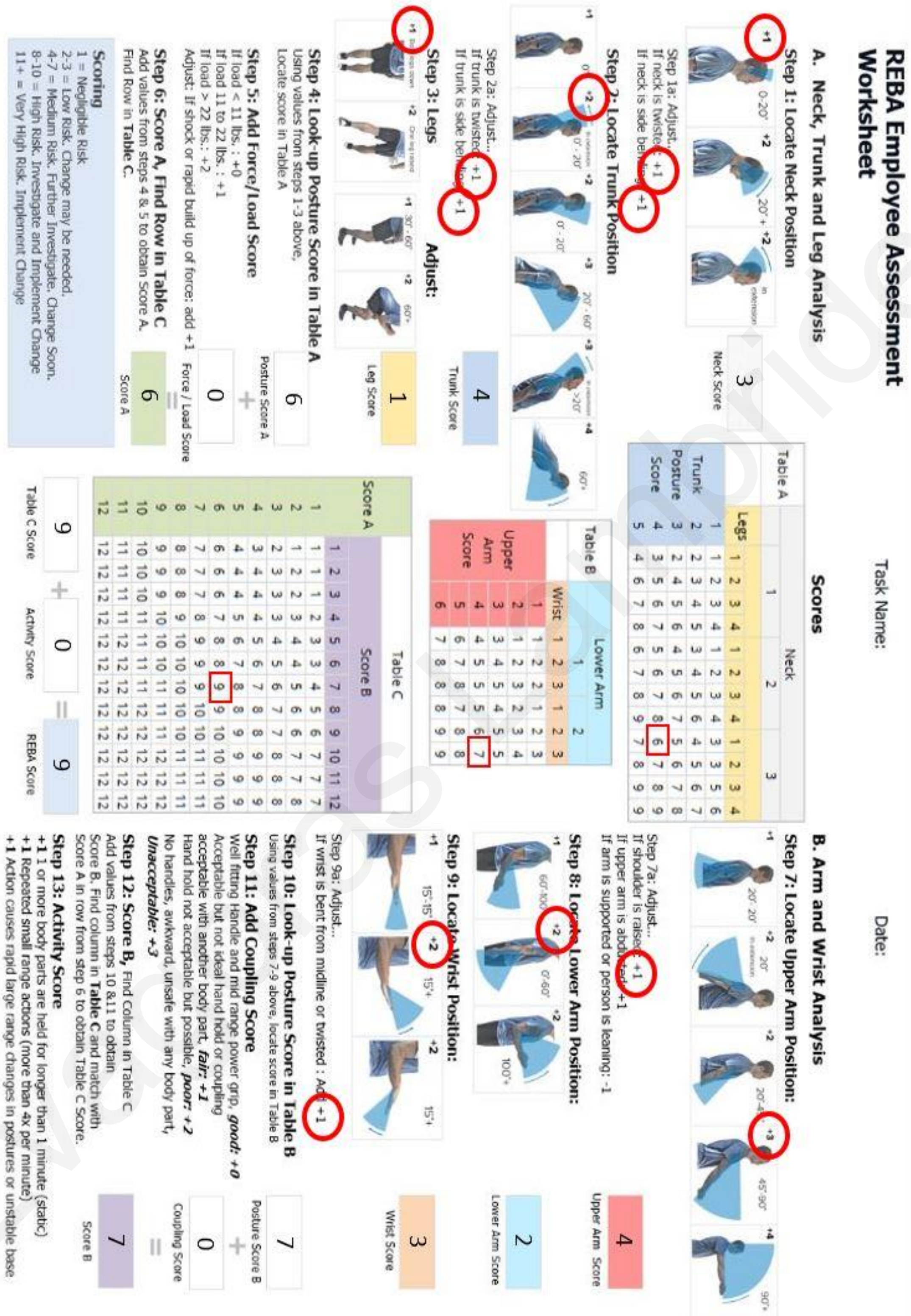


Figure 4.6: Pose 3 Manual results

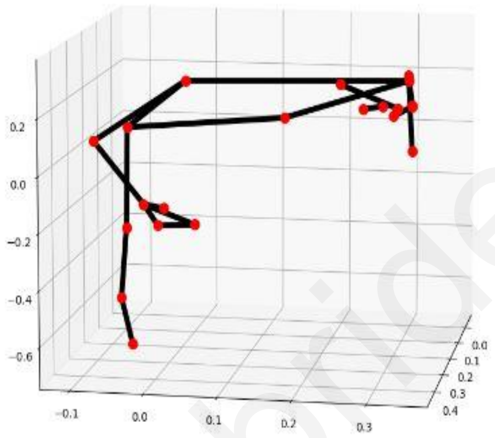
Discussion:

In pose 3 we again have a difference of 2 points between the manual solution and the code, with the total scores being equal to ‘9’ and ‘11’ respectively. As in pose 2, there is an error regarding the pose estimation, since the right leg of the worker does not appear in the figure extracted by the code and the left arm is overextended. It is also observed that the left leg is shown up to the knee. For these reasons, we get an incorrect leg score (+4) from the code, while in the manual solution we have a score equal to 1, which corresponds better to the specific pose. The neck position is found at the limit of 0 to 20 degrees, it is twisted and side-bent and therefore it is graded with ‘+3’.

As for the trunk position, it is rated ‘+4’ since it is tilted backwards, is twisted and side bending. There is a difference of one point with the code rating (+5). The scores related to the arms are consistent between the code and the manual method, with the upper arm getting 4 points (45-90 deg and raised shoulder), the lower arm 2 points (0-60 deg) and the wrist 3 points (15+ deg and bent from midline).

POSE 4:

(a)



(b)

Figure 4.7: Before and after pose-estimation analysis: (a) Worker pose 4, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.4: Pose 4 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">6</td><td colspan="3">4</td><td>7</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">Medium Risk. Further Investigate. Change Soon</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>3</td><td>3</td><td>2</td><td>2</td><td>2</td><td>3</td></tr>
</table>

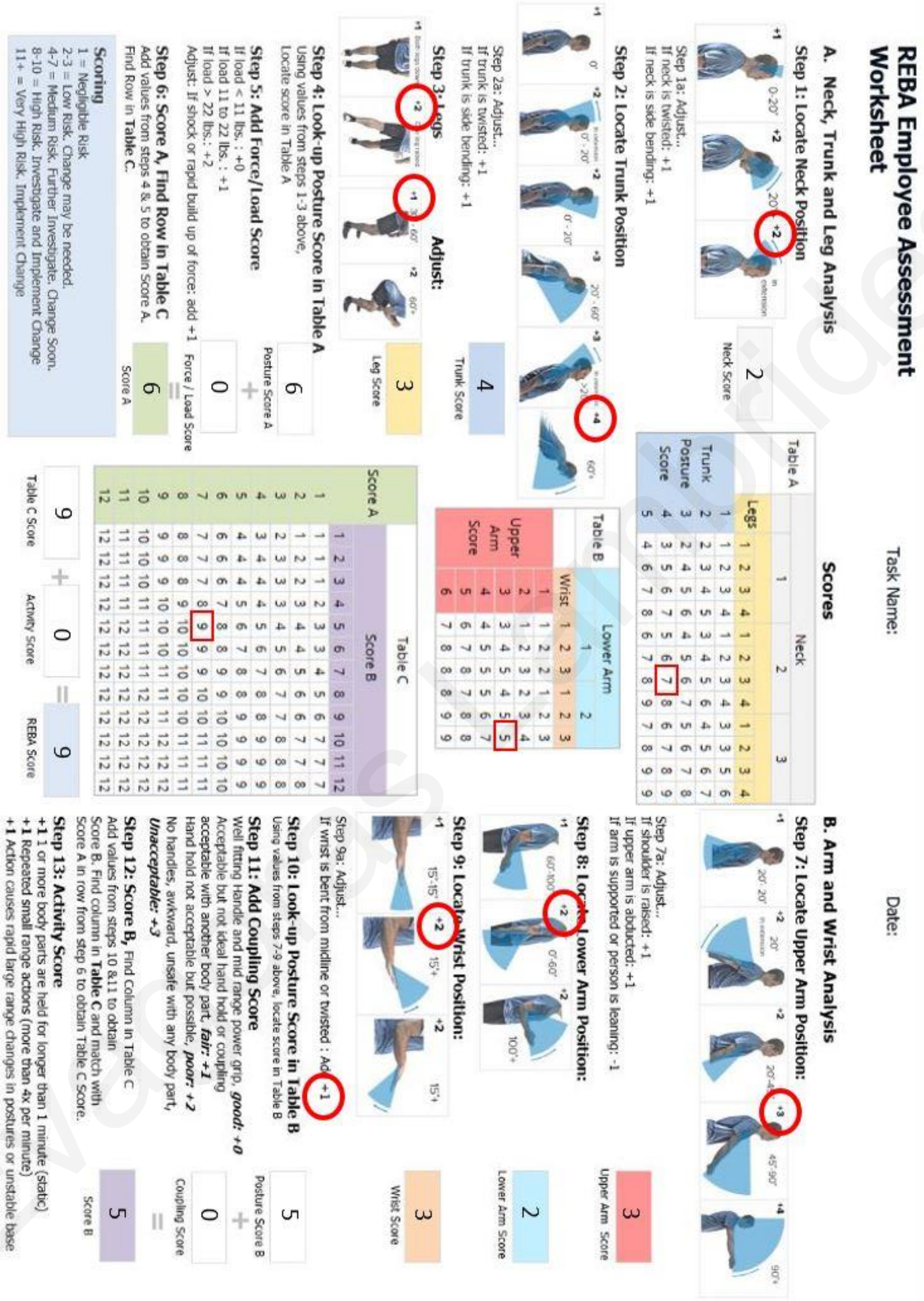
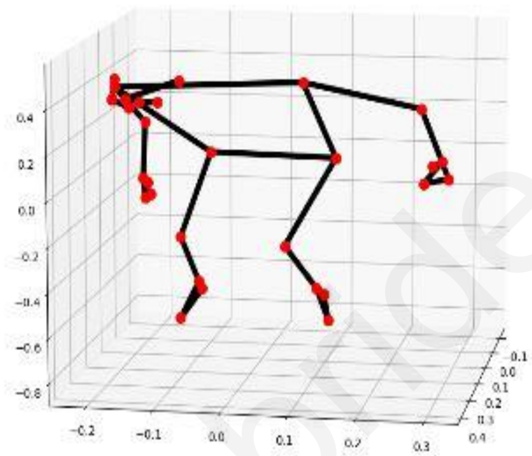


Figure 4.8: Pose 4 Manual results

Discussion:

In pose 4 we have yet another case of erroneous pose estimation by the software code since, as we can see from the result, the worker's right leg and hand are not correctly recognized from the photo resulting in several differences in the individual REBA scores. From the two methods we get a total REBA Score of 9 (manual) and 7 (code), with most of the differences being in Part A (Neck, Trunk, Legs). The inclination of the neck is backward without being side bending or twisted (+2) and the trunk position is estimated to exceed 60 degrees on forward. The legs are in unilateral weight bearing and with an inclination between 30 and 60 degrees on one leg (+3). Next, the upper arm is graded with +3 (45-90 degrees), the lower arm with +2 (0-60 degrees) and the wrist with +3 (15+ and bend from midline). The difference here was the upper arm which was rated by the code with '2'.

POSE 5:



(a) (b)

Figure 4.9: Before and after pose-estimation analysis: (a) Worker pose 5, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.5: Pose 5 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">8</td><td colspan="3">4</td><td>9</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">High Risk. Investigate and Implement Change</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>3</td><td>3</td><td>4</td><td>2</td><td>2</td><td>3</td></tr>
</table>

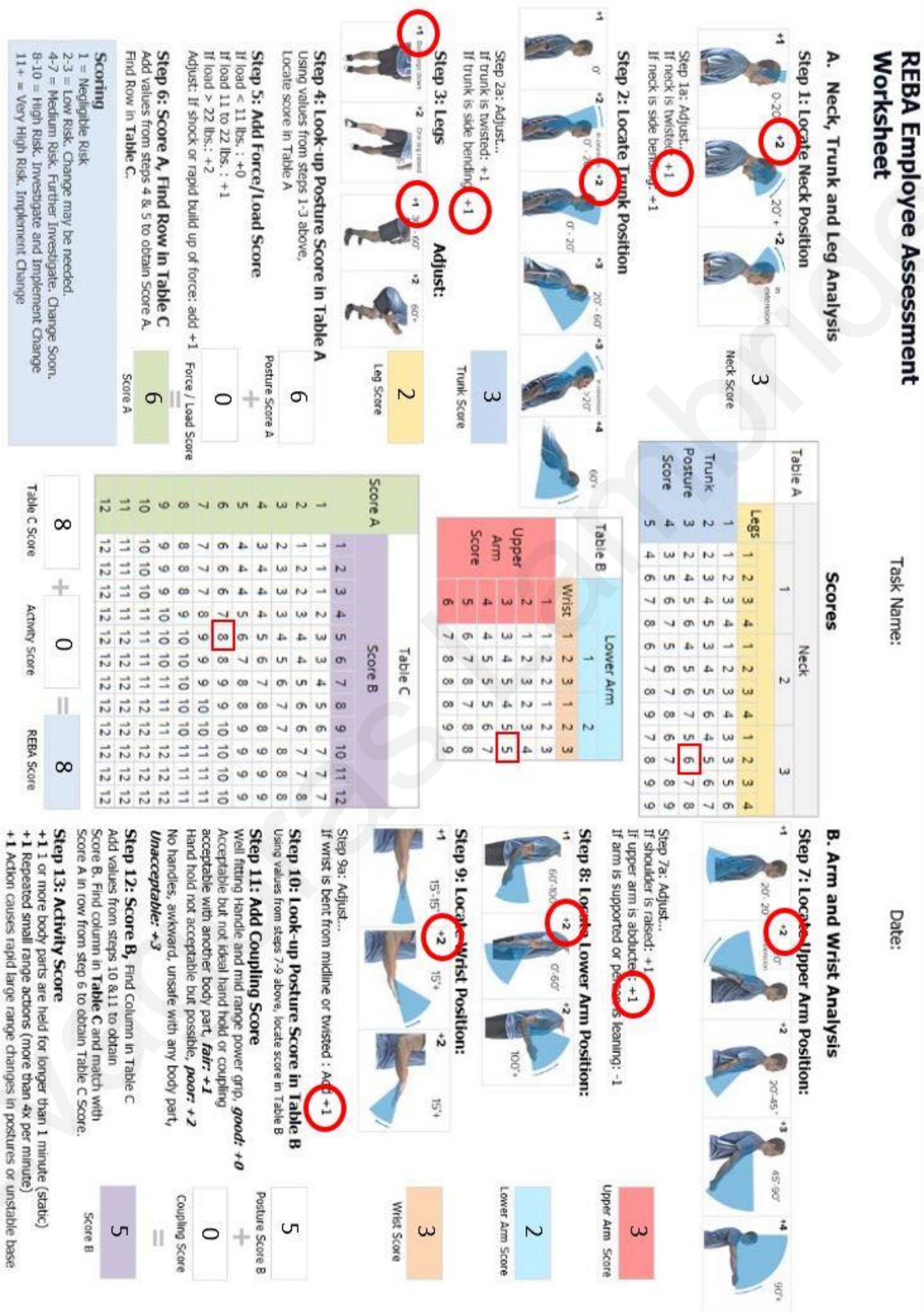
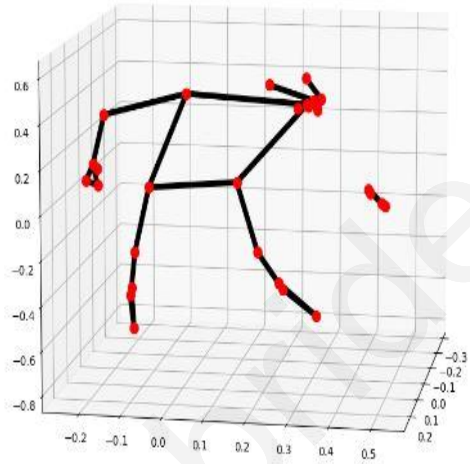


Figure 4.10: Pose 5 Manual results

Discussion:

At first look of the results of pose 5, it is observed that the biggest difference between the two methods is in the legs score, with the code giving a score equal to ‘4’ and the manual solution ‘2’. In this case we have an obvious bilateral weight bearing between the legs and the inclination of the knees does not exceed 60 degrees. As for the neck (20+ degrees and twisted) and trunk (0-20 degrees and side bending) positions, are rated with +3 in both cases, while the Lower Arm (0-60 degrees) and wrist (15+ degrees and twisted) are rated with 2 and 3 respectively. The upper arm obviously has a backward tilt and is also abducted, which is why its graded with a ‘3’ in the manual solution. However, the code evaluates it with ‘2’. In total, in this pose we have a REBA score equal to 9 for the program and equal to 8 for the manual solution.

POSE 6:



(a) (b)

Figure 4.11: Before and after pose-estimation analysis: (a) Worker pose 6, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.6: Pose 6 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">8</td><td colspan="3">5</td><td>10</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">High Risk. Investigate and Implement Change</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>3</td><td>5</td><td>2</td><td>3</td><td>2</td><td>3</td></tr>
</table>

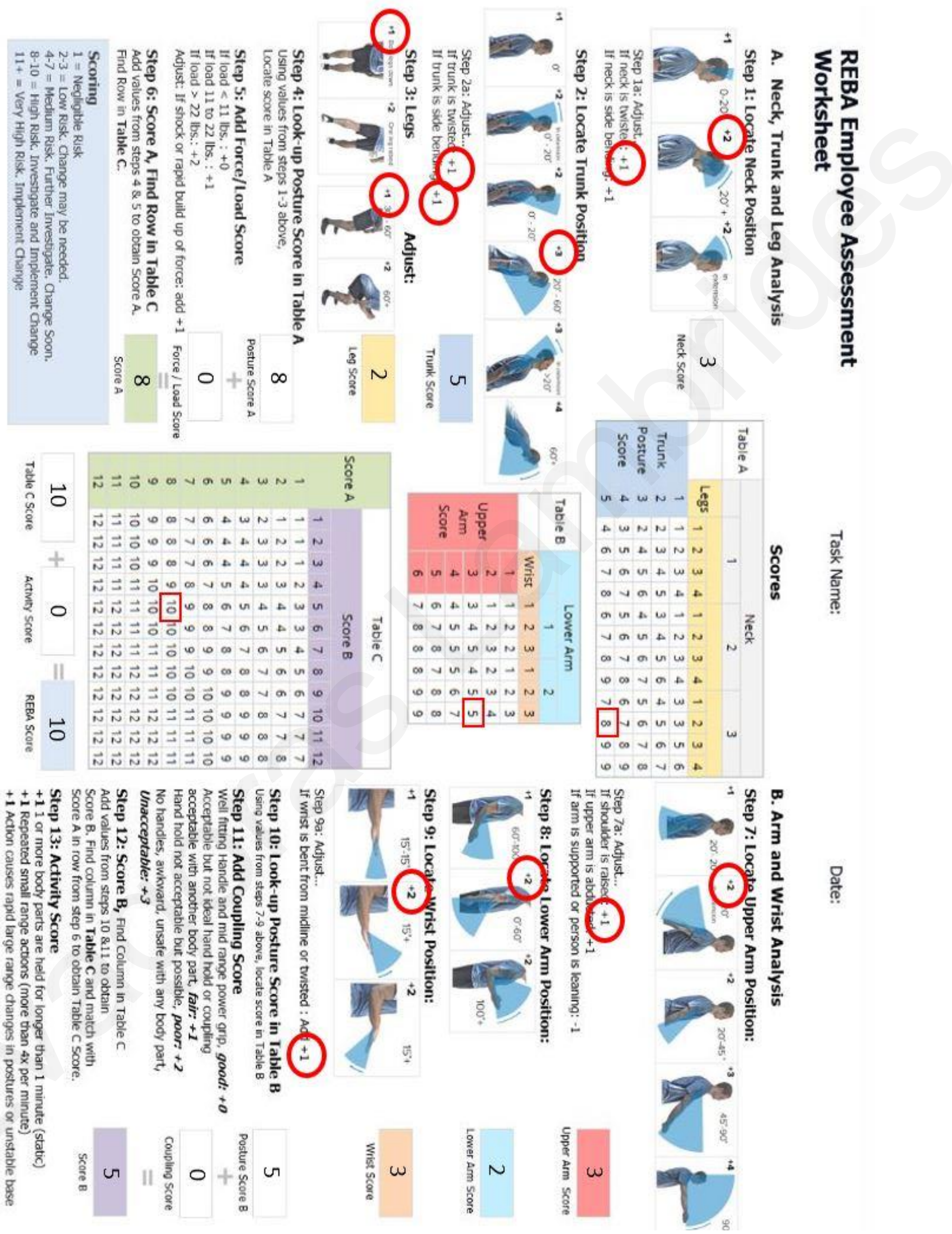


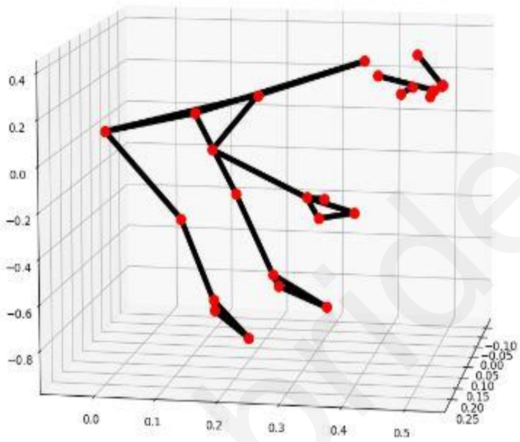
Figure 4.12: Pose 6 Manual results

Discussionn:

The case of Pose 6 shows, as before, an error in the pose estimation process. As a result, part of the right hand does not appear in the skeletonized form of the pose. Despite this error the results between the code and the manual solver are completely consistent with a final Reba Score equal to 10. Starting from the neck position we have 20+ degrees of forward inclination and the neck is twisted, while the trunk position is between 20 and 60 degrees (+3) and is also twisted and side bending (+2). The legs have bilateral weight bearing and an inclination at the knees approaching 30 degrees (+2). Moving on to Part b which concern the arms, the upper arm is tilted back, and the shoulder is raised (+2), while the position of the lower arm is at the limit of 0-60 degrees (+2). Finally, the wrist is graded with a maximum score of +3 since it is at an inclination of 15+ degrees and is also twisted.

POSE 7:

(a)



(b)

Figure 4.13: Before and after pose-estimation analysis: (a) Worker pose 7, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.7: Pose 7 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">8</td><td colspan="3">5</td><td>10</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">High Risk. Investigate and Implement Change</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>3</td><td>5</td><td>2</td><td>3</td><td>2</td><td>3</td></tr>
</table>

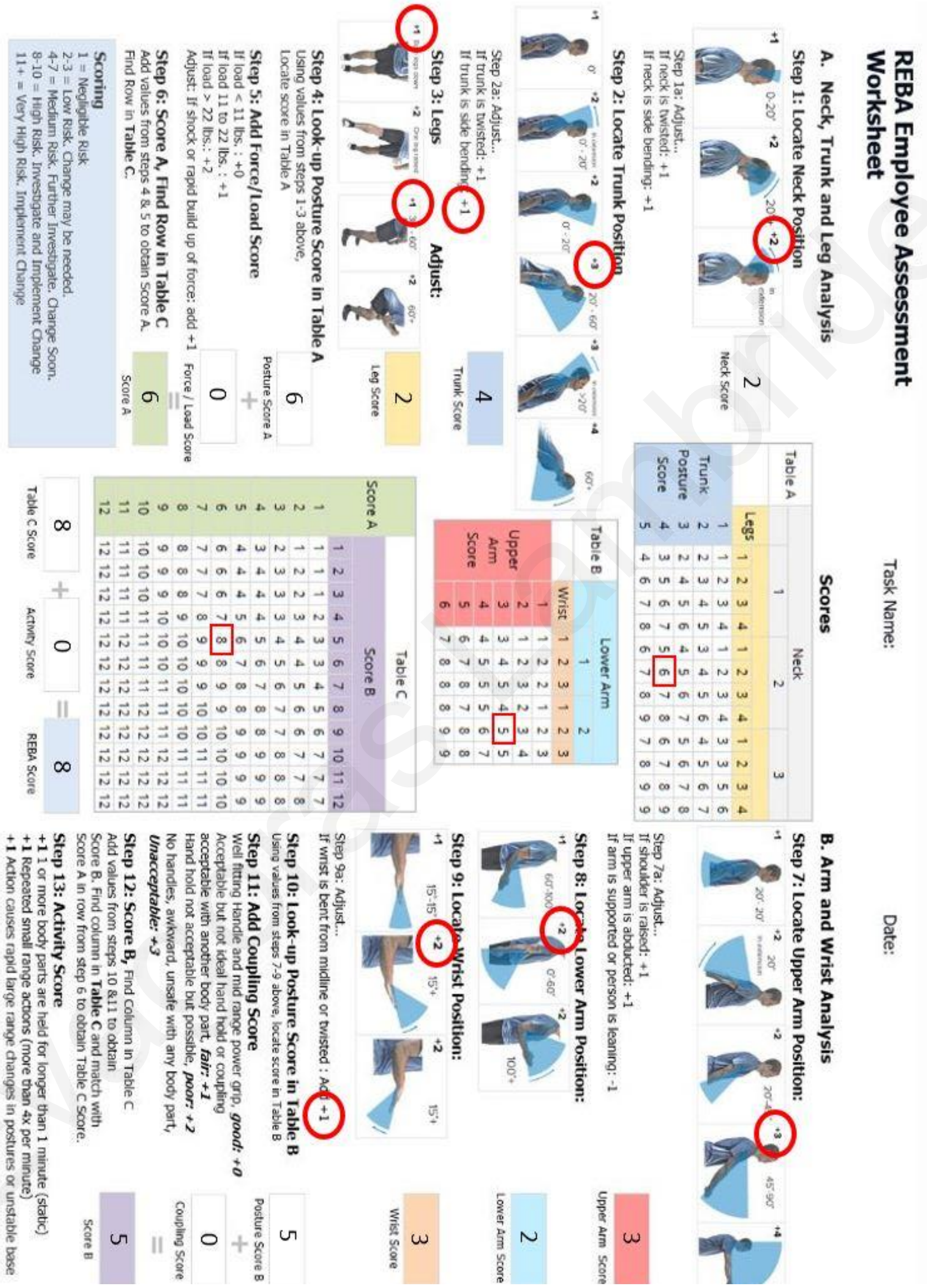
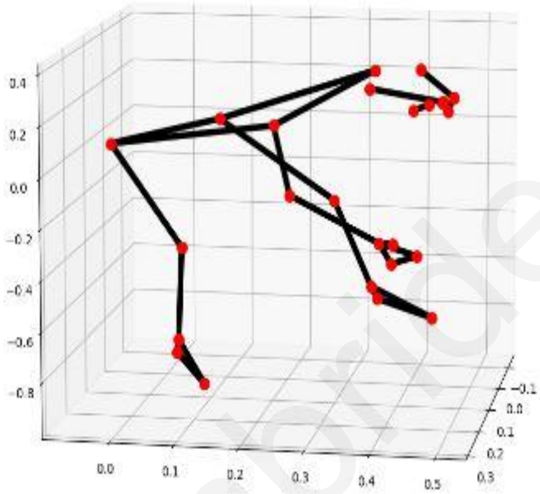


Figure 4.14: Pose 7 Manual results

Discussion:

In pose 7, we observe the absence of the left hand from the skeletonized form of the pose but, in this case, we cannot refer to an error since in this photo capture only the right hand is visible. The final scores differ by 2 points ('8' manual and '10' code) with the partial differences focusing exclusively on Part A and the Neck, Trunk, Legs analysis. Starting from the neck position, in the manual solution we have +2 points for tilting the neck backwards, while in code it is scored with 3. The trunk position is estimated to have an inclination between 20 and 60 degrees and is side bending so it has a score of +4 (+5 in code). The manual score for the legs is consistent with the code and equals 2 with unilateral weight bearing and knee flexion between 30-60 degrees. Regarding the arms, the upper arm is between 45-90 degrees (+3), the lower arm is around 50 degrees (+2) and the wrist is bent 15+ degrees upwards and is also bend from midline (+3).

POSE 8:



(a) (b)

Figure 4.15: Before and after pose-estimation analysis: (a) Worker pose 8, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.8: Pose 8 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">7</td><td colspan="3">4</td><td>8</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">High Risk. Investigate and Implement Change</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>2</td><td>5</td><td>2</td><td>2</td><td>2</td><td>3</td></tr>
</table>

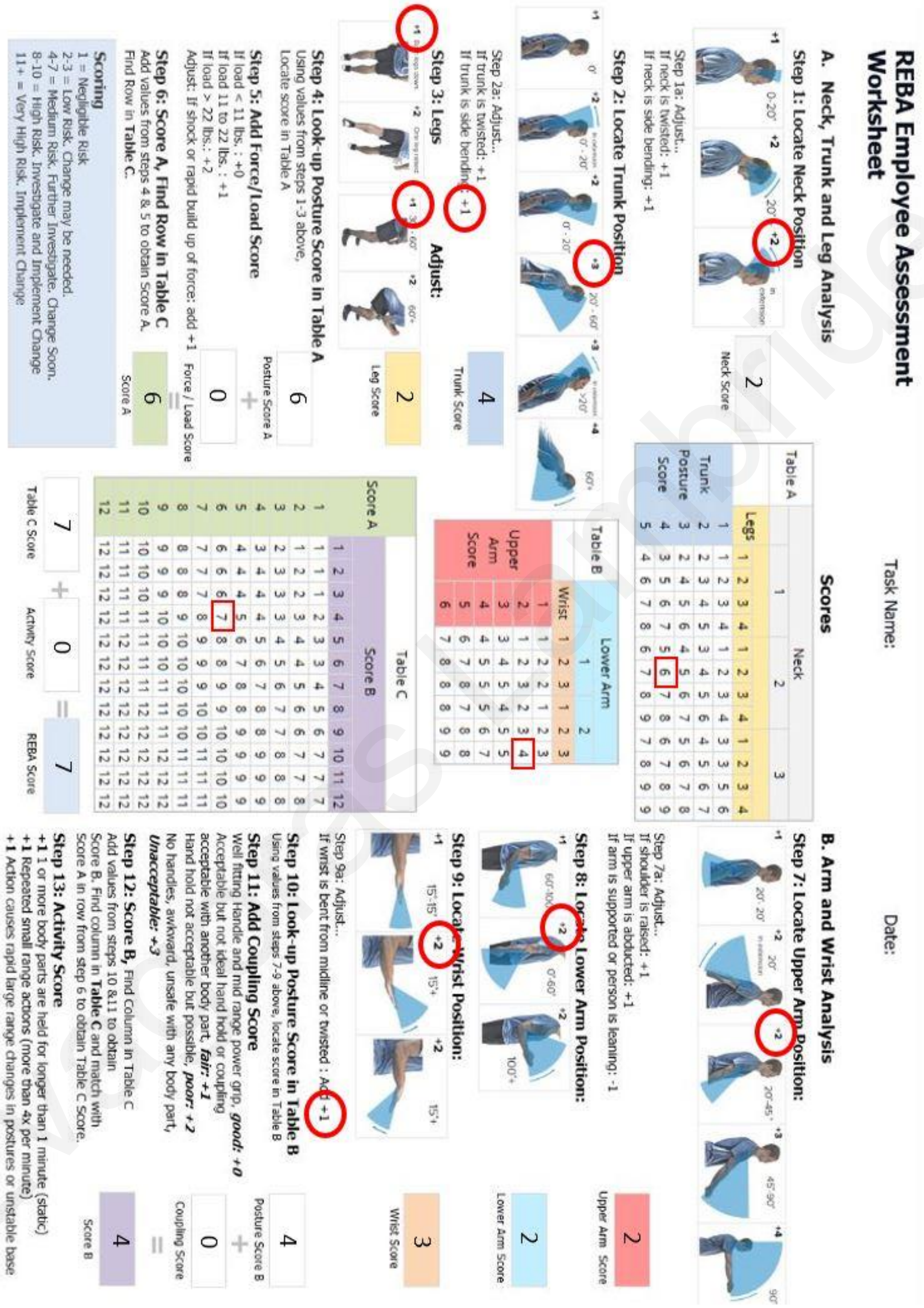
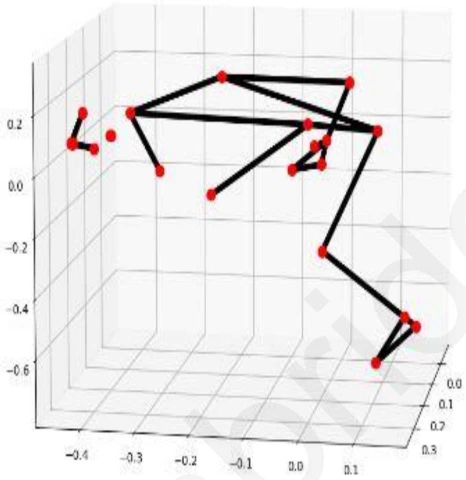


Figure 4.16: Pose 8 Manual results

Discussion:

Moving on to pose 8, we have a total score of ‘8’ and ‘7’ for code and manual solution respectively, with the only difference between the two methods being in the trunk position score. In part A the neck position gets 2 points since it is tilted back and the legs are also scored +2 considering that they are in unilateral weight bearing (+1) and are tilted (30-60 degrees). In the manual method the trunk position was rated +4 for 20-60 degrees forward tilt (+3) and side bending (+1). In the code we get a score equal to 5. Regarding Part b and the arms, in the upper arm we have a slope of 20-45 degrees (+2) and in the lower arm a slope of 0-60 degrees (+2). Finally, the wrist with 3 points has an inclination of 15+ degrees and is bent from midline.

POSE 9:



(a) (b)

Figure 4.17: Before and after pose-estimation analysis: (a) Worker pose 9, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.9: Pose 9 Code results

SCORE A			SCORE B			SCORE C
9			**5**			**10**
Partial			Partial			High Risk. Investigate and Implement Change
Neck	Trunk	Legs	Upper Arm	Lower Arm	Wrist	
3	4	4	3	2	3	

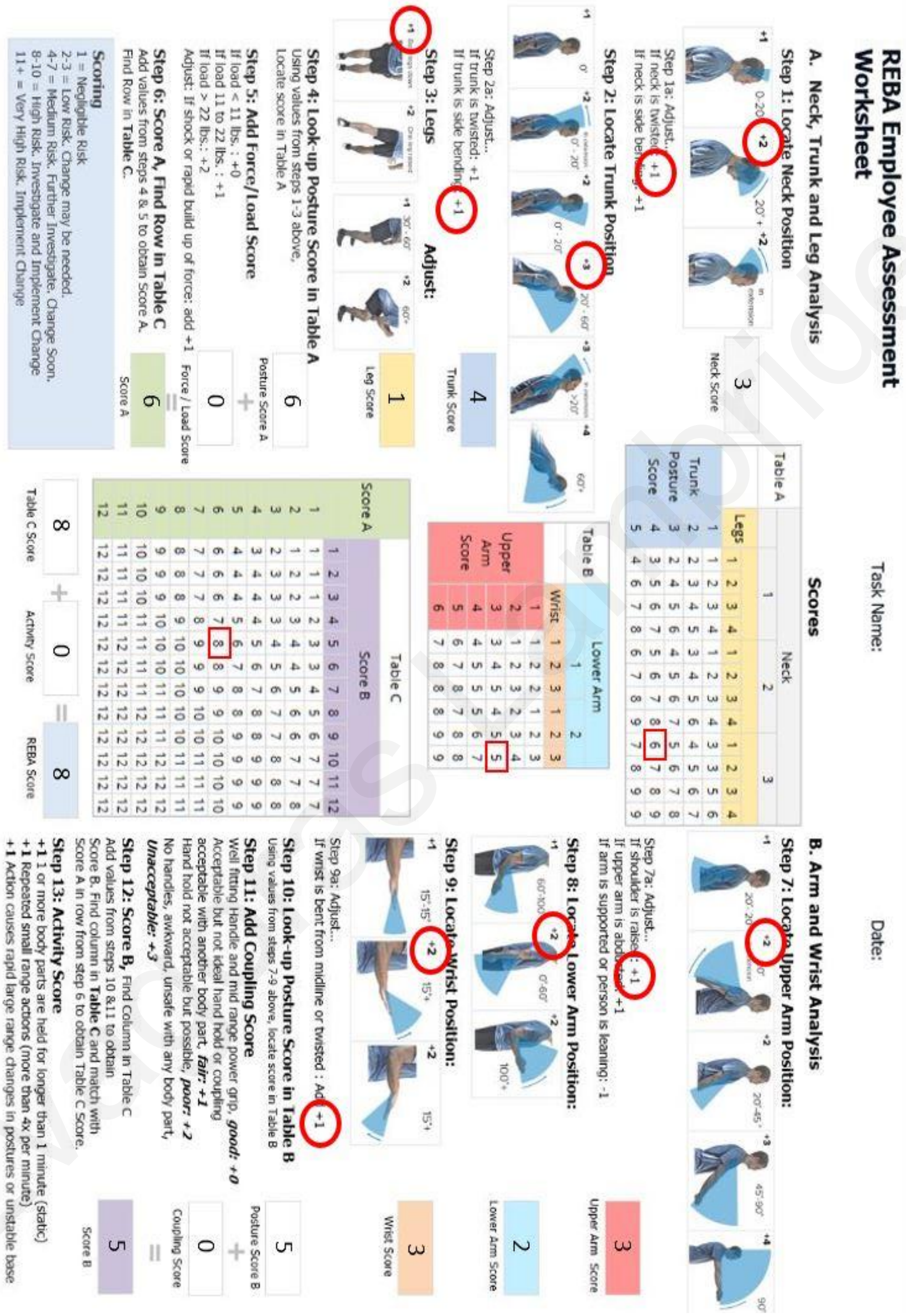


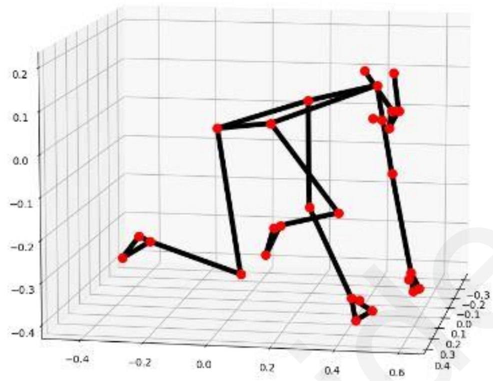
Figure 4.18: Pose 9 Manual results

Discussion:

Pose 9 is another case of error in the pose estimation phase. The issue is found on one of the two legs of the figure. Therefore, a large discrepancy is observed in the score concerning the legs, with the program score being 4 while in the manual method is 1. This is the only difference in the individual scores which, however, causes a difference of 2 points in the overall score (code score=10, manual score=8). The neck position is found at a slope greater than 20 degrees and the neck is twisted (+3). Trunk position varies between 20 – 60 degrees and is side bending at the same time. Moving on to the hands we have +3 points for the upper arm which has a negative slope backwards and the shoulder is raised. The lower arm ranges from 0 – 60 degrees (+2), while the wrist tilts more than 15 degrees upwards and is also twisted.

POSE 10:

(a)



(b)

Figure 4.19: Before and after pose-estimation analysis: (a) Worker pose 10, (b) Skeletonized pose showing body part keypoints.

Code Results:

Table 4.10: Pose 10 Code results

<table>
<tr><th colspan="3">SCORE A</th><th colspan="3">SCORE B</th><th>SCORE C</th></tr>
<tr><td colspan="3">5</td><td colspan="3">3</td><td>4</td></tr>
<tr><td colspan="3">Partial</td><td colspan="3">Partial</td><td rowspan="3">Medium Risk. Further Investigate. Change Soon</td></tr>
<tr><td>Neck</td><td>Trunk</td><td>Legs</td><td>Upper Arm</td><td>Lower Arm</td><td>Wrist</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>2</td><td>1</td><td>3</td></tr>
</table>

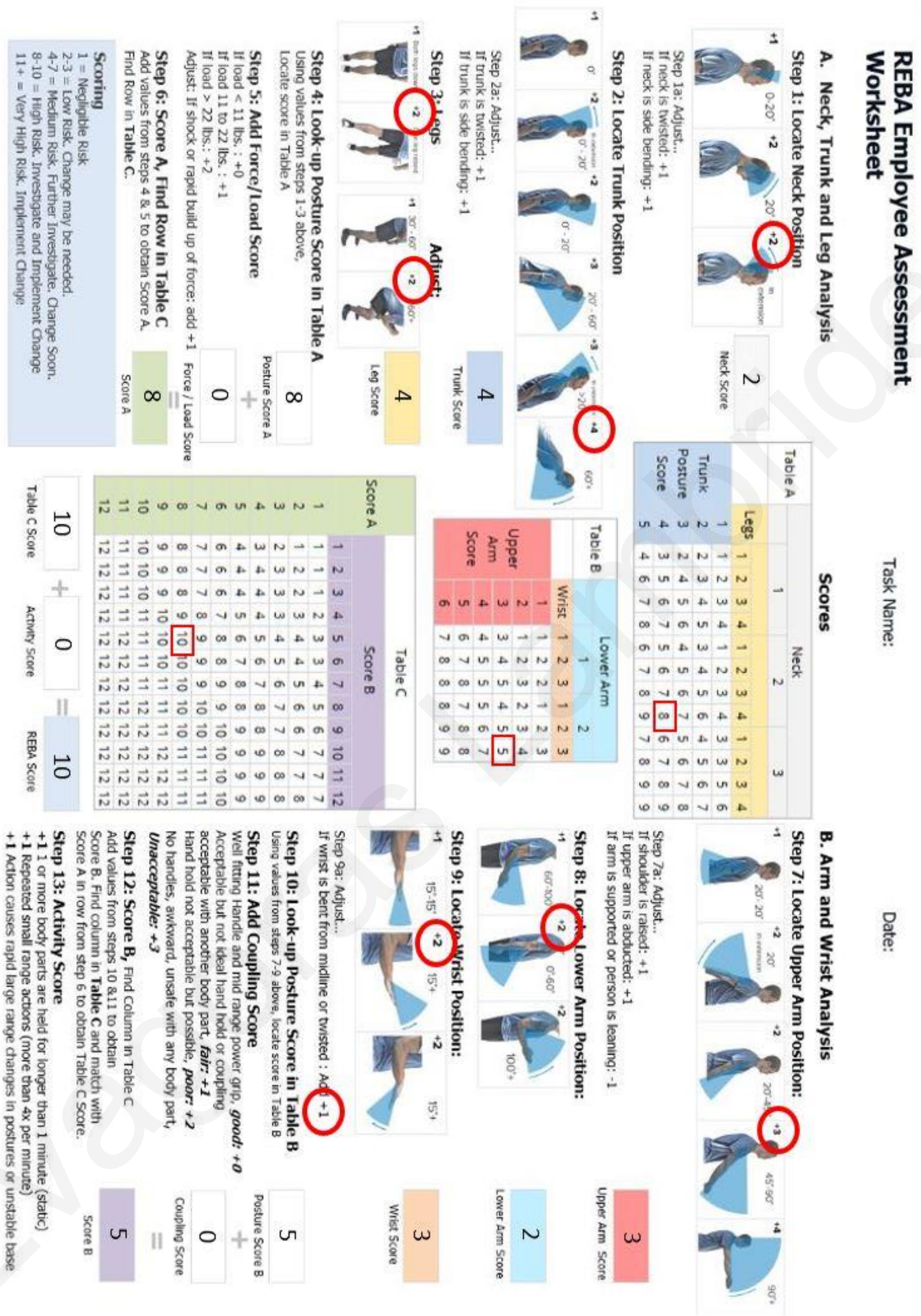


Figure 4.20: Pose 10 Manual results

Discussion:

Pose 10 is the last case studied in which we find the largest difference in total scores between the two methods than all the rest of the poses. Specifically, the code score is equal to 4 while the manual score has a difference of 6 points and is equal to 10. Starting again from the neck position, we have a negative slope and a score of +2, which together with the wrist (+3 for 15+ degrees of inclination and bent from midline) are the only partial scores that are consistent between the two methods. The trunk is obviously inclined more than 60 degrees and for this reason it gets 4 points as opposed to the code which evaluates it with 3. As for the legs we have unilateral weight bearing (+2) and an inclination at the knees that clearly exceeds the 30 degrees (+2) which gives us leg score equal to 4, greater than that of the code (+2). Finally, the position of the upper arm varies between 45-90 degrees (+3) and the lower arm between 0-60 degrees (+2). The corresponding scores of the code are +2 and +1.

5. SUMMARY OF FINDINGS

Studying the results extracted from the analyses of the 10 figures discussed in the previous section, we are led to certain conclusions and observations regarding the effectiveness, reliability and the way of application for the two methods (code, manual).

Initially, one of the main factors that is taken into consideration in the practical application of REBA is the speed with which the analysis of the pose will be done in each case that will be evaluated, with the code obviously superior to the manual method. The speed of application and the simplicity provided by the code is also its biggest advantage, since the method saves time, effort and cost to the contractor through the automation of the whole process.

Subsequently, the results allow us to make a complete evaluation of the efficiency and reliability of the code by studying any obvious errors that either concern the pose estimation or the REBA scores. In most examples analyzed above, we notice some differences in the overall and individual scores (REBA) between the two methods as well as, less often, in the pose estimation. This does not necessarily refer to a code error since the results of the manual method are derived from the evaluator's personal assessment of body posture and therefore its results cannot be considered completely correct. These two cases (pose estimation and REBA scores) are interrelated since a possibly incorrect estimation of the code in the pose estimation will lead to an incorrect calculation of slopes between the body parts of the figure and therefore to an incorrect REBA score. A typical example of pose estimation error is the case of Pose 4 in which the figure's left arm and leg are not shown in the skeletonized form. This type of error (pose estimation) is due in most cases to the "unfavourable" capture of the photo which can hide some body parts or even to the quality of the image. Regarding the errors related to the scores, pose 10 is the most extreme example since it has the biggest difference (6 pt) in the overall score (REBA score) from all the other poses. Observing the posture of the body we easily conclude that in this case it is clearly a wrong evaluation of the code. On the other hand, there are several examples in which the scores between the two methods are very close, with a range of +/-1 or even completely consistent with each other. Typical are the cases of Pose 6 and Pose 8 with scores 10 and 8 respectively.

Achieving a full 3D visualization of the human pose is a serious challenge for the CV method since every different caption received as input to the software must be evaluated so that 3D coordinates for each body part can be extracted from a 2D image. This constitutes the greatest difficulty of the analysis and at the same time testifies to the ability of the CV method to gain understanding about the environment it "sees" in order to make a decision. Also, as mentioned above, any obstructions of body parts in an image (input) can easily affect the accuracy of the method, with the results of the analysis lack of the coordinates for the hidden body parts or even in rarer cases to recognize the objects that hide the human as human body parts. This issue might be solved by using video instead of a picture, so that through movement there is a better view of the human pose and an improved perception of space by the software.

In general, the code can be considered reliable and workable, with the exception of 1 out of 10 examples studied, and the results are reasonable and expected. Errors are presented but not to the extent that they could affect the reliability of the method. REBA is impractical to be implemented manually by humans because one cannot constantly monitor the movements of the workers, while using a mobile or fixed camera inside the construction site would be much more practical and easier.

6. CONCLUSIONS

Upgrading the methods of compliance and implementation of the health and safety plan is an urgent in our time, and the development of technology provides us with the possibility to carry it out successfully. In the present study, i focused on the part that concerns body posture during work at the construction site, and its evaluation using the REBA method. The whole process was automated by applying it to a CV-based method, using code (Python), and from which, the results were extracted and studied.

The ultimate goal of the project was the study of a method which, with its application in the construction industry, would significantly contribute to the minimization of WMSDs which is a long-term problem presented to people engaged in construction works. WMSDs and injuries can cause workers pain and suffering and even loss of income if they become unable to work. This fact also creates significant costs for employers which may include reduced productivity and increased workers' compensation insurance premiums.

In the project, 10 photos of workers performing typical work at a construction site were used as data and analysed with the REBA method. The analysis was done using CV (code) and manually, giving us the possibility for a comparison between the results. Through the process that was carried out, the appropriate conclusions were exported regarding the effectiveness, reliability and application of the two methods, with CV being superior since it provides greater convenience, speed and is more applicable to the construction site, which is what is requested.

The execution of construction projects always presupposes a safe workforce, and the use of CV techniques in the field of civil engineering must be perceived as a key part to improve monitoring of health and safety conditions.

The general idea is to create a healthier and more progressive environment by using appropriate safety and health techniques during the construction of projects which will significantly contribute to reduce injuries within the construction site, as well as health problems (WMSD), that may be caused in long term.

REFERENCES

[1] "Eurostat Website." Nov. 20, 2022. [Online]. Available: https://ec.europa.eu/
[2] "Forbes Website." Nov. 20, 2022. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=4dbb40b51018
[3] "IBM (International Business Machines) Website." Nov. 20, 2022. [Online]. Available: https://www.ibm.com/us-en/
[4] "viAct Website." Nov. 20, 2022. [Online]. Available: https://www.viact.ai/post/top-7-computer-vision-applications-for-construction
[5] "DeepX Website." Nov. 20, 2022. [Online]. Available: https://deepxhub.com/
[6] M. F. Antwi-Afari, H. Li, D. J. Edwards, E. A. Pärn, J. Seo, and A. Y. L. Wong, "Biomechanical analysis of risk factors for work-related musculoskeletal disorders during repetitive lifting task in construction workers," *Automation in Construction*, vol. 83, pp. 41–47, Nov. 2017, doi: 10.1016/j.autcon.2017.07.007.
[7] "CPWR (The Center for Construction Research and Training) Website." Nov. 20, 2022. [Online]. Available: https://www.cpwr.com/
[8] "Velocity EHS Website." Nov. 20, 2022. [Online]. Available: https://www.ehs.com/2020/05/how-to-use-the-rapid-entire-body-assessment/
[9] "ErgoPlus Website." Nov. 20, 2022. [Online]. Available: https://ergo-plus.com/reba-assessment-tool-guide/#:~:text=The%20REBA%20score%20represents%20the,is%20a%20score%20of%204
[10] "Physiopedia Website." Nov. 20, 2022. [Online]. Available: https://www.physio-pedia.com/Rapid_Entire_Body_Assessment_(REBA)
[11] S. Hignett and L. McAtamney, "Rapid Entire Body Assessment (REBA)," *Applied Ergonomics*, vol. 31, no. 2, pp. 201–205, Apr. 2000, doi: 10.1016/S0003-6870(99)00039-3.
[12] "LearnOpenCV Website." Nov. 20, 2022. [Online]. Available: https://learnopencv.com/building-a-body-posture-analysis-system-using-mediapipe/
[13] "MediaPipe Pose Estimation (Python)." Nov. 20, 2022. [Online]. Available: https://google.github.io/mediapipe/solutions/pose.html
[14] "Reba Score calculation." Nov. 20, 2022. [Online]. Available: https://github.com/rs9000/ergonomics

APPENDIX

Python code for pose estimation

```
import cv2
import mediapipe as mp
import time
from datetime import datetime as dt
import numpy as np


mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_holistic = mp.solutions.holistic



#----------------------------------------------------------------------
# Get a screenshot of the screen (or a named window) and save it as
# an image file
#----------------------------------------------------------------------
def screenshot():
    from subprocess import call
    import pyautogui

    #pyautogui.getWindowsWithTitle("Figure 1")[0].maximize()
    myScreenshot = pyautogui.screenshot()
    myScreenshot.save('screenshot.png')

    ## for macOS
    #call(["screencapture", "-i", "/screenshot.jpg"])

#----------------------------------------------------------------------
# show multiple images into a 2x2 image wall
#----------------------------------------------------------------------
def img_wall(imgfile1, imgfile2, imgfile3, imgfile4):
    import cv2
    import numpy as np

    # Read Images
    img1 = cv2.imread(imgfile1)
    img2 = cv2.imread(imgfile2)
    img3 = cv2.imread(imgfile3)
    img4 = cv2.imread(imgfile4)
    # concatenate image Horizontally
    Hori = np.concatenate((img1, img2), axis=1)

    # concatanate image Vertically
    Verti = np.concatenate((img3, img4), axis=0)

    cv2.imshow('HORIZONTAL', Hori)
    cv2.imshow('VERTICAL', Verti)
```

```
    cv2.waitKey(0)
    cv2.destroyAllWindows()

#----------------------------------------------------------------------
# convert mediapipe body parts/pairs to reba-compatible body
# parts/pairs
#----------------------------------------------------------------------
def mp_to_reba(file):
    # mediapipe's pose features
    # 0: nose                   10: mouth_right            20:
right_index         30: right_heel
    # 1: left_eye_inner         11: left_shoulder          21:
left_thumb          31: left_foot_index
    # 2: left_eye               12: right_shoulder         22:
right_thumb         32: right_foot_index
    # 3: left_eye_outer         13: left_elbow             23:
left_hip
    # 4: right_eye_inner        14: right_elbow            24:
right_hip
    # 5: right_eye              15: left_wrist             25:
left_knee
    # 6: right_eye_outer        16: right_wrist            26:
right_knee
    # 7: left_ear               17: left_pinky             27:
left_ankle
    # 8: right_ear              18: right_pinky            28:
right_ankle
    # 9: mouth_left             19: left_index             29:
left_heel
    #
    lst_mp_parts_all = ['nose', 'left_eye_inner', 'left_eye',
'left_eye_outer', 'right_eye_inner', 'right_eye',
                 'right_eye_outer', 'left_ear', 'right_ear',
'mouth_left', 'mouth_right',
                 'left_shoulder', 'right_shoulder', 'left_elbow',
'right_elbow', 'left_wrist',
                 'right_wrist', 'left_pinky', 'right_pinky',
'left_index', 'right_index',
                 'left_thumb', 'right_thumb', 'left_hip',
'right_hip', 'left_knee',
                 'right_knee', 'left_ankle', 'right_ankle',
'left_heel', 'right_heel',
                 'left_foot_index', 'right_foot_index'
                    ]
    # REBA ergonomics's pose features
    # The input pose is a 13 + 2 joints representing X, Y, Z
coordinates relative to the root joint.
    # [0] = Head
    # [1] = Nose
    # [2, 3, 4, 14]: Left Shoulder, Elbow, Wrist + Hand(optional)
    # [5, 6, 7, 15]: Right Shoulder, Elbow, Wrist + Hand(optional)
    # [8, 9, 10]: Left Hip, Knee, Ankle
    # [11, 12, 13]: Right Hip, Knee, Ankle

    # sample_pose = np.array([
    #      [ 0.08533354,  1.03611605,  0.09013124],
    #      [ 0.15391247,  0.91162637, -0.00353906],
    #      [ 0.22379057,  0.87361878,  0.11541229],
```

```
    #      [ 0.4084777 ,  0.69462843,  0.1775224 ],
    #      [ 0.31665226,  0.46389668,  0.16556387],
    #      [ 0.1239769 ,  0.82994377, -0.11715403],
    #      [ 0.08302169,  0.58146328, -0.19830338],
    #      [-0.06767788,  0.53928527, -0.00511249],
    #      [ 0.11368726,  0.49372503,  0.21275574],
    #      [ 0.069179  ,  0.07140968,  0.26841402],
    #      [ 0.10831762, -0.36339359,  0.34032449],
    #      [ 0.11368726,  0.41275504, -0.01171348],
    #      [ 0.        ,  0.        ,  0.        ],
    #      [ 0.02535541, -0.43954643,  0.04373671],
    #      [ 0.26709431,  0.33643749,  0.17985192],
    #      [-0.15117603,  0.49462711,  0.02703403]])

    print('Time : ', dt.fromtimestamp(curr_time).strftime('%Y-%m-%d
%H:%M:%S'))

    # coordinates of joints, BEFORE re-referencing
    print('-----------------------------------------------------------')
    print('coordinates of joints, BEFORE re-referencing ...\n')
    for idx, item in enumerate(lst_mp_parts_all):
        # print(item, " at index ", idx)
        coord = results.pose_landmarks.landmark[idx]
        print(''.join([str(idx), ': ', item, ' @ (', str(coord.x),
',', str(coord.y), ',', str(coord.z), ')']))

    # convert to REBA-compliant coordinates: (RHip + LHip)/2 is set
as the center of origin (0,0,0) and all
    # features are positioned in relation to that point
    poi_x =
(results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_HIP].x
+

results.pose_landmarks.landmark[mp_holistic.PoseLandmark.RIGHT_HIP].x
) / 2
    poi_y =
(results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_HIP].y
+

results.pose_landmarks.landmark[mp_holistic.PoseLandmark.RIGHT_HIP].y
) / 2
    poi_z =
(results.pose_landmarks.landmark[mp_holistic.PoseLandmark.LEFT_HIP].z
+

results.pose_landmarks.landmark[mp_holistic.PoseLandmark.RIGHT_HIP].z
) / 2

    # poi = np.array((poi_x, poi_y, poi_z))

    ## coordinates of ALL joints, AFTER re-referencing w.r.t. hips
    #for idx, item in enumerate(lst_mp_parts):
    #    # print(item, " at index ", idx)
    #    coord = results.pose_landmarks.landmark[idx]
    #    print(''.join([str(idx), ': ', item, ' @ (', str(coord.x -
poi_x), ',', str(coord.y - poi_y), ',', str(coord.z - poi_z), ')']))
```

```
    # coordinates of REBA joints of interest, AFTER re-referencing
w.r.t. hips
    # REBA                                  MediaPipe
    # 0 : Head                              1 : left_eye_inner (??)
    # 1 : Nose                              0 : nose
    # 2 : LShoulder                         11: left_shoulder
    # 3 : LElbow                            13: left_elbow
    # 4 : LWrist                            15: left_wrist
    # 14: LHand (optional)                  19: left_index (??)
    # 5 : RShoulder                         12: right_shoulder
    # 6 : RElbow                            14: right_elbow
    # 7 : RWrist                            16: right_wrist
    # 15: RHand(optional)                   20: right_index (??)
    #  8: LHip                              23: left_hip
    #  9: LKnee                             25: left_knee
    # 10: LAnkle                            27: left_ankle
    # 11: RHip                              24: right_hip
    # 12: RKnee                             26: right_knee
    # 13: RAnkle                            28: right_ankle


    lst_mp_parts_selected = ['left_eye_inner', 'nose',
                             'left_shoulder', 'left_elbow',
'left_wrist', 'left_index',
                             'right_shoulder', 'right_elbow',
'right_wrist', 'right_index',
                             'left_hip', 'left_knee', 'left_ankle',
                             'right_hip', 'right_knee', 'right_ankle'
                             ]

    print('--------------------------------------------------------')
    print('coordinates of joints, AFTER re-referencing and re-sorted
to match REBA analysis ... \n')
    body_angles = []
    for idx, item in enumerate(lst_mp_parts_selected):
        # print(item, " at index ", idx)
        coord = results.pose_landmarks.landmark[idx]
        print(''.join([str(idx), ': ', item, ' @ (', str(coord.x -
poi_x), ',', str(coord.y - poi_y), ',', str(coord.z - poi_z), ')']))

        lst_temp = [[round(coord.x - poi_x,6), round(coord.y -
poi_y,6), round(coord.z - poi_z,6)]]
        body_angles.extend(lst_temp)

    #output angles to a text file
    filename = os.path.basename(file).split('.')[0]
    path = os.getcwd() + '/' + filename
    with open(path + "/" + filename + '_pose.txt', 'w') as f:
        f.write("[\n")
        for item in body_angles[:-1]:           # loop through all
list items (except last one)
            f.write("%s, \n" % item)            # add a comma at the
end of them
        for item in body_angles[-1:]:           # add the last item
            f.write("%s \n" % item)            # without a comma at
the end of it
        f.write("]\n")
```

```
#-----------------------------------------------------------------------
# image input
#-----------------------------------------------------------------------
import os
from os import listdir
from os.path import isfile, join
from PIL import ImageGrab           # to capture an image from screen



# utils for drawing on image
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_pose = mp.solutions.pose

# get all jpg images in current directory, to analyze
IMAGE_FILES = []
IMAGE_FILES = [f for f in os.listdir(os.getcwd()) if
f.endswith('construction22.jpg')]


BG_COLOR = (192, 192, 192) # gray
with mp_pose.Pose(
    static_image_mode=True,
    enable_segmentation=True,
    min_detection_confidence=0.5) as pose:
  for idx, file in enumerate(IMAGE_FILES):
      print('Image processing ...', file)
      # create a subdirectory (if id doesnt exist) to store processed
files
      filename = os.path.basename(file).split('.')[0]
      # Check whether the specified path exists or not
      path= os.getcwd() + '/' + filename
      isExist = os.path.exists(path)
      if not isExist:
          os.makedirs(path)


      #proceed with the image analysis
      image = cv2.imread(file)
      cv2.imwrite(path + "/" + filename + '_orig.png', image)
      image_height, image_width, _ = image.shape
      # Convert the BGR image to RGB before processing.
      results = pose.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))

      if not results.pose_landmarks:
          print('pose landmarks not found')
          continue

      #print(
      #    f'Nose coordinates: ('
      #
f'{results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE].x *
image_width}, '
```

```
        #
f'{results.pose_landmarks.landmark[mp_pose.PoseLandmark.NOSE].y *
image_height})'
        #)

        annotated_image = image.copy()
        # Draw segmentation on the image.
        # To improve segmentation around boundaries, consider applying
a joint
        # bilateral filter to "results.segmentation_mask" with "image".

        condition = np.stack((results.segmentation_mask,) * 3, axis=-1)
> 0.1
        bg_image = np.zeros(image.shape, dtype=np.uint8)
        bg_image[:] = BG_COLOR
        annotated_image = np.where(condition, annotated_image,
bg_image)
        # Draw pose landmarks on the image.
        mp_drawing.draw_landmarks(
            annotated_image,
            results.pose_landmarks,
            mp_pose.POSE_CONNECTIONS,

landmark_drawing_spec=mp_drawing_styles.get_default_pose_landmarks_st
yle())

        cv2.imwrite(path + "/" + filename + '_annotated.png',
annotated_image)

        # Plot pose world landmarks
        mp_drawing.plot_landmarks(
            results.pose_world_landmarks, mp_pose.POSE_CONNECTIONS)

        #screenshot()

        # convert the detected pose to a REBA-compliant model; passing
the filename as an argument
        if results.pose_landmarks:
            # convert the detected pose to a reba-specific pose (
[(LHip + RHip)/2] defines point 0,0,0)
            curr_time = time.time()  # grab the current time
            mp_to_reba(file)

        #show detected pose
        image = cv2.imread(filename + '_annotated.png')
        cv2.imshow("Pose Detection - ", image)
        if cv2.waitKey(5) & 0xFF == 27:
            break

        # show all images in an image wall
        img_wall(
            file,                              # original image
            filename + '_annotated.png',       # annotated image
            file,
            filename + '_annotated.png'
        )
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()


#-----------------------------------------------------------------------
# webcam input
#-----------------------------------------------------------------------

cap = cv2.VideoCapture(0)
last_recorded_time = time.time() # this keeps track of the last time
a frame was processed

with mp_holistic.Holistic(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as holistic:
  while cap.isOpened():
    success, image = cap.read()
    img_orig = image
    curr_time = time.time()  # grab the current time
    if not success:
      print("Ignoring empty camera frame.")
      # If loading a video, use 'break' instead of 'continue'.
      continue

    # To improve performance, optionally mark the image as not
writeable to
    # pass by reference.
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    results = holistic.process(image)

    # Draw landmark annotation on the image.
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    image_height, image_width, _ = image.shape

    mp_drawing.draw_landmarks(
        image,
        results.face_landmarks,
        mp_holistic.FACEMESH_CONTOURS,
        landmark_drawing_spec=None,
        connection_drawing_spec=mp_drawing_styles
        .get_default_face_mesh_contours_style())
    mp_drawing.draw_landmarks(
        image,
        results.pose_landmarks,
        mp_holistic.POSE_CONNECTIONS,
        landmark_drawing_spec=mp_drawing_styles
        .get_default_pose_landmarks_style())


    # export to file
    filename =
'image'+dt.fromtimestamp(curr_time).strftime('%Y%m%d_%H%M%S')
    #img_blank = np.zeros((image_height, image_width, 3), np.uint8)
# create a blank image, to save the mediapipe skeleton
```

```
    cv2.imwrite(filename + '_wiSk.png', image)    # with skeleton
    # cv2.imwrite(filename + '_woSk.png', img_orig)    # without
skeleton
    img_OnlySkeleton = cv2.subtract(image, img_orig)        #just
skeleton
    cv2.imwrite(filename + '_juSk.png', img_OnlySkeleton)  # just
skeleton


    # extract the x,y,z coordinates of specific body parts
    #x_coordinate =
results.pose_landmarks.landmark[mp_holistic.PoseLandmark.NOSE].x *
image_width
    #y_coordinate =
results.pose_landmarks.landmark[mp_holistic.PoseLandmark.NOSE].y *
image_height

    # in the cycle of cam-capture, check if a feature exists and
extract its coordinates
    # 0: nose                   10: mouth_right             20:
right_index         30: right_heel
    # 1: left_eye_inner         11: left_shoulder           21:
left_thumb          31: left_foot_index
    # 2: left_eye               12: right_shoulder          22:
right_thumb         32: right_foot_index
    # 3: left_eye_outer         13: left_elbow              23:
left_hip
    # 4: right_eye_inner        14: right_elbow             24:
right_hip
    # 5: right_eye              15: left_wrist              25:
left_knee
    # 6: right_eye_outer        16: right_wrist             26:
right_knee
    # 7: left_ear               17: left_pinky              27:
left_ankle
    # 8: right_ear              18: right_pinky             28:
right_ankle
    # 9: mouth_left             19: left_index              29:
left_heel

    # output at set intervals (say 1 seconds) the detected human pose
features
    if results.pose_landmarks and (curr_time - last_recorded_time >=
1.0): # it has been at least 1 seconds:
        # convert the detected pose to a reba-specific pose ( [(LHip
+ RHip)/2] defines point 0,0,0)
        mp_to_reba()
        # reset last_recorded_time
        last_recorded_time = curr_time


    # Flip the image horizontally for a selfie-view display.
    cv2.imshow('MediaPipe Holistic', cv2.flip(image, 1))
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
```

Python libraries used in code (‘requirements.txt’):

```
absl-py==1.0.0
attrs==21.2.0
cycler==0.11.0
DateTime==4.3
kiwisolver==1.3.2
matplotlib==3.4.3
mediapipe==0.8.9
MouseInfo==0.1.3
numpy==1.21.4
opencv-contrib-python==4.5.4.58
opencv-python==4.5.4.58
Pillow==8.4.0
protobuf==3.19.1
PyAutoGUI==0.9.53
PyGetWindow==0.0.9
PyMsgBox==1.0.9
pyobjc==8.2
pyobjc-core==8.2
pyobjc-framework-Accessibility==8.2
pyobjc-framework-Accounts==8.2
pyobjc-framework-AddressBook==8.2
pyobjc-framework-AdServices==8.2
pyobjc-framework-AdSupport==8.2
pyobjc-framework-AppleScriptKit==8.2
pyobjc-framework-AppleScriptObjC==8.2
pyobjc-framework-ApplicationServices==8.2
pyobjc-framework-AppTrackingTransparency==8.2
pyobjc-framework-AudioVideoBridging==8.2
pyobjc-framework-AuthenticationServices==8.2
pyobjc-framework-AutomaticAssessmentConfiguration==8.2
pyobjc-framework-Automator==8.2
pyobjc-framework-AVFoundation==8.2
pyobjc-framework-AVKit==8.2
pyobjc-framework-BusinessChat==8.2
pyobjc-framework-CalendarStore==8.2
pyobjc-framework-CallKit==8.2
pyobjc-framework-CFNetwork==8.2
pyobjc-framework-ClassKit==8.2
pyobjc-framework-CloudKit==8.2
pyobjc-framework-Cocoa==8.2
pyobjc-framework-Collaboration==8.2
pyobjc-framework-ColorSync==8.2
pyobjc-framework-Contacts==8.2
pyobjc-framework-ContactsUI==8.2
pyobjc-framework-CoreAudio==8.2
```

pyobjc-framework-CoreAudioKit==8.2
pyobjc-framework-CoreBluetooth==8.2
pyobjc-framework-CoreData==8.2
pyobjc-framework-CoreHaptics==8.2
pyobjc-framework-CoreLocation==8.2
pyobjc-framework-CoreMedia==8.2
pyobjc-framework-CoreMediaIO==8.2
pyobjc-framework-CoreMIDI==8.2
pyobjc-framework-CoreML==8.2
pyobjc-framework-CoreMotion==8.2
pyobjc-framework-CoreServices==8.2
pyobjc-framework-CoreSpotlight==8.2
pyobjc-framework-CoreText==8.2
pyobjc-framework-CoreWLAN==8.2
pyobjc-framework-CryptoTokenKit==8.2
pyobjc-framework-DeviceCheck==8.2
pyobjc-framework-DictionaryServices==8.2
pyobjc-framework-DiscRecording==8.2
pyobjc-framework-DiscRecordingUI==8.2
pyobjc-framework-DiskArbitration==8.2
pyobjc-framework-DVDPlayback==8.2
pyobjc-framework-EventKit==8.2
pyobjc-framework-ExceptionHandling==8.2
pyobjc-framework-ExecutionPolicy==8.2
pyobjc-framework-ExternalAccessory==8.2
pyobjc-framework-FileProvider==8.2
pyobjc-framework-FileProviderUI==8.2
pyobjc-framework-FinderSync==8.2
pyobjc-framework-FSEvents==8.2
pyobjc-framework-GameCenter==8.2
pyobjc-framework-GameController==8.2
pyobjc-framework-GameKit==8.2
pyobjc-framework-GameplayKit==8.2
pyobjc-framework-ImageCaptureCore==8.2
pyobjc-framework-IMServicePlugIn==8.2
pyobjc-framework-InputMethodKit==8.2
pyobjc-framework-InstallerPlugins==8.2
pyobjc-framework-InstantMessage==8.2
pyobjc-framework-Intents==8.2
pyobjc-framework-IOSurface==8.2
pyobjc-framework-iTunesLibrary==8.2
pyobjc-framework-KernelManagement==8.2
pyobjc-framework-LatentSemanticMapping==8.2
pyobjc-framework-LaunchServices==8.2
pyobjc-framework-libdispatch==8.2
pyobjc-framework-LinkPresentation==8.2
pyobjc-framework-LocalAuthentication==8.2
pyobjc-framework-MapKit==8.2

pyobjc-framework-MediaAccessibility==8.2
pyobjc-framework-MediaLibrary==8.2
pyobjc-framework-MediaPlayer==8.2
pyobjc-framework-MediaToolbox==8.2
pyobjc-framework-Metal==8.2
pyobjc-framework-MetalKit==8.2
pyobjc-framework-MetalPerformanceShaders==8.2
pyobjc-framework-MetalPerformanceShadersGraph==8.2
pyobjc-framework-MLCompute==8.2
pyobjc-framework-ModelIO==8.2
pyobjc-framework-MultipeerConnectivity==8.2
pyobjc-framework-NaturalLanguage==8.2
pyobjc-framework-NetFS==8.2
pyobjc-framework-Network==8.2
pyobjc-framework-NetworkExtension==8.2
pyobjc-framework-NotificationCenter==8.2
pyobjc-framework-OpenDirectory==8.2
pyobjc-framework-OSAKit==8.2
pyobjc-framework-OSLog==8.2
pyobjc-framework-PassKit==8.2
pyobjc-framework-PencilKit==8.2
pyobjc-framework-Photos==8.2
pyobjc-framework-PhotosUI==8.2
pyobjc-framework-PreferencePanes==8.2
pyobjc-framework-PushKit==8.2
pyobjc-framework-Quartz==8.2
pyobjc-framework-QuickLookThumbnailing==8.2
pyobjc-framework-ReplayKit==8.2
pyobjc-framework-SafariServices==8.2
pyobjc-framework-SceneKit==8.2
pyobjc-framework-ScreenSaver==8.2
pyobjc-framework-ScreenTime==8.2
pyobjc-framework-ScriptingBridge==8.2
pyobjc-framework-SearchKit==8.2
pyobjc-framework-Security==8.2
pyobjc-framework-SecurityFoundation==8.2
pyobjc-framework-SecurityInterface==8.2
pyobjc-framework-ServiceManagement==8.2
pyobjc-framework-Social==8.2
pyobjc-framework-SoundAnalysis==8.2
pyobjc-framework-Speech==8.2
pyobjc-framework-SpriteKit==8.2
pyobjc-framework-StoreKit==8.2
pyobjc-framework-SyncServices==8.2
pyobjc-framework-SystemConfiguration==8.2
pyobjc-framework-SystemExtensions==8.2
pyobjc-framework-UniformTypeIdentifiers==8.2
pyobjc-framework-UserNotifications==8.2

pyobjc-framework-UserNotificationsUI==8.2
pyobjc-framework-VideoSubscriberAccount==8.2
pyobjc-framework-VideoToolbox==8.2
pyobjc-framework-Virtualization==8.2
pyobjc-framework-Vision==8.2
pyobjc-framework-WebKit==8.2
pyparsing==3.0.6
pyperclip==1.8.2
PyRect==0.1.4
PyScreeze==0.1.28
python-dateutil==2.8.2
pytweening==1.0.4
pytz==2021.3
rubicon-objc==0.4.2
six==1.16.0
zope.interface==5.4.0