ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Human Activity Recognition with Deep Learning

By

Dimitris Chortarias

Submitted

in partial fulfilment of the requirements for the degree of

Master of Big Data and Analytics

at the

UNIVERSITY OF PIRAEUS

Month 2021

Author ....

MSc “Big Data and Analytics”

Month 09,2021

Certified by ....

Magklogiannis Ilias

Professor

Thesis Supervisor

Certified by ....

Filippakis Michael

Associate Professor

Member of Examination

Committee

Certified by ....

Kyriazis Dimosthenis

Associate Professor

Member of Examination

Committee

Human Activity Recognition with Deep Learning

By

Dimitris Chortarias

Submitted to the MSc "Big Data and Analytics" on Month 00, 2021, in

partial fulfillment of the requirements for the MSc degree

Abstract

Human activity recognition is one of the biggest challenges in the Computer Vision era that is coming these years, including Virtual Reality and Augmented Reality. Computer vision is a broad term for the work done with deep neural networks to develop human-like vision capabilities for applications. It can include specific training of neural nets for segmentation, classification and detection using images and videos for data.

In this master thesis, there have been created 4 videos from real world practicing for the muscles of shoulder and with the use of Computer Vision try to check and give feedback if the gym exercise has been done correctly or not. To achieve the best outcome for this it has been created and using multiple advanced data modelling techniques.

Thesis Supervisor:

Title:

Acknowledgments

First and foremost, I am very grateful to Prof. Ilias Maglogiannis of the Dept. of Digital Systems of the University of Piraeus for supervising my research and providing me the guidance and necessary resources needed for completing this thesis.

I would also like to express my gratitude to my parents and friends for their tremendous support, understanding and encouragement in the past year.

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the view of the University of Piraeus.

Contents

List of Figures

List of Tables

1.Introduction

In this section is mentioned the importance of computer vision based on deep learning techniques for helping people with their workouts.

1.1 Motivation

In the past one and a half year due to Covid restrictions and while this paper was written the gyms all over the country have been closed. So, many people in order to keep their fitness continue their workout schedules at home. However, there is often one problem with these types of exercises; there is no one to watch and correct the movements of the body. This can lead human body to fatal errors for the muscles and can create huge orthopedic problems. Also, during this time new companies as start-ups have been initiated in order to have indoor bike and virtual trainer video in order to give better instructions. In the category of muscles and training with dumbbells the cost of the trainer is the same with having a personal trainer. On the other side without covid restrictions, you are never sure that the trainer in the gym is watching if you are performing the right moves of the workout. At all this time, many of the sites are motivating people to stay fit with some online videos. [1] [2] Many often we are having a subscription in the gyms selecting them according to the annual fees, but if the annual fees are lower also the trainers that are in the gym are less.This is not good for people's fitness but helps the gym in business plans and business targets. Due to all these above and while after the covid era we will try to protect ourselves more, computer vision is the way to keep our fitness and be sure that we train our bodies with the maximum performance that we are able to.

1.2 Deep Learning for Human Activity Recognition

Using Computer Vision and Deep Learning to gain data and the Human pose estimation from each frame of the video remains a challenge. One video can be in many resolutions from 320x640 to 1080x720, it can depend on the angle from which the video is taken or if there is enough light to the key points of the body like feet or hands. In many videos and based in the angle taken the key points of the body maybe are not visible to a frame of the video. The data that we need to use from the video due to high resolution 1080x720 many times is going to show as the center of the hand in an area of ten pixels next to the previous frame. All these are some of the challenges that we are facing in this thesis.

One of the questions is why we are using computer vision now. In the previous years we were not able to do all described above. this is because of the faster CPU and GPU that our computers nowadays are using. The data for a simple point of the body if we are going to have this data to an excel file for a video that lasts 10 seconds with FPS (frames per second) 120 is going to take about 1200 rows and if we need the 5 body limbs head, hands and feet these data are going to be 5 times more and this means 6000 rows for a simple 10 second video. Imagine running a deep learning 12000 total frames and 5 body limps, in the computers of past years this can take 5 to 10 minutes, but with the newest CPU and GPU this can be done in milliseconds.

1.3 Structure of thesis

This thesis is subdivided in the following different topics:

Chapter 1: Introduction refers to the topic and highlights the scope and objectives of the present thesis.

Chapter 2: Brief Introduction to Machine learning and Deep Learning.

Chapter 3: Related work.

Chapter 4: The implementation of our work.

Chapter 5: Conclusion and future work is a summary of the accomplished work as well as a brief exploration of future research opportunities. Bibliography provides a list of sources referred to this thesis, to further facilitate reader’s access to the selected articles and books.

2. Brief Introduction to Machine learning and Deep Learning.

In this section there is a brief introduction to machine learning. As mentioned before there is the approach that the part of machine learning that the model has used is named Computer Vision, which is one of the many fields of machine learning and deep learning.

2.2 Machine learning

We can define machine learning as a set of methods that can identify and find patterns in our data and we can apply these methods in order to predict the future or classify our data in segments. There are three main types of machine learning Supervised learning, Unsupervised learning and Reinforcement Learning. The three types of machine learning can split into many other categories and in the next sections there is a summary of these after a quick brief on machine learning history.

2.2.1 The history of machine learning

Many books and researchers appear to have a different opinion about when was the first implementation in machine learning. Although we need to agree that the simplest way of y=ax+b is just a simple linear regression where a+b are the bias which the model need to apply which is more than 200 years old. Some books focus on the first neutral network from Frank Rosenblatt's Perceptron.

A psychologist, Rosenblatt conceived of the Percetron as a simplified mathematical model of how the neurons in our brains operate: it takes a set of binary inputs (nearby neurons), multiplies each input by a continuous valued weight (the synapse strength to each nearby neuron), and thresholds the sum of these weighted inputs to output a 1 if the sum is big enough and otherwise a 0 (in the same way neurons either fire or do not). Most of the inputs to a Perceptron are either some data or the output of another Perceptron, but an extra detail is that Perceptrons also have one special 'bias' input, which has just a value of 1 and basically ensures that more functions are computable with the same input by being able to offset the summed value. This model of the neuron built on the work of Warren McCulloch and Walter Pitts Mcculoch-Pitts, who showed that a neuron model that sums binary inputs and outputs a 1 if the sum exceeds a certain threshold value, and otherwise outputs a 0, can model the basic OR/AND/NOT functions. This, in the early days of AI, was a big deal - the predominant thought at the time was that making computers able to perform formal logical reasoning would essentially solve AI.

On the other hand, at the paper On the Origin of Deep Learning from Haohan Wang and Bhiksha Raj, they are focused on Aristotle from 300 BC with the Contribution of introduced Associationism, started the history of human's attempt to understand brain.

Table 1: The History of Machine learning

Year	Contributer	Contribution
300 BC	Aristotle	introduced Associationism, started the history of human's attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model.
1949	Donald Hebb	considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1958	Frank Rosenblatt	introduced the first perceptron, which highly resembles modern perceptron
1974	Paul Werbos	introduced Backpropagation
1980	Teuvo Kohonen	introduced Self Organizing Map
	Kunihiko Fukushima	introduced Neocogitron, which inspired Convolutional Neural Network
1982	John Hopfield	introduced Hopfield Network
1985	Hilton & Sejnowski	introduced Boltzmann Machine
1986	Paul Smolensky	introduced Harmonium, which is later known as Restricted Boltzmann Machine
	Michael I. Jordan	defined and introduced Recurrent Neural Network
1990	Yann LeCun	introduced LeNet, showed the possibility of deep neural networks in practice
1997	Schuster & Paliwal	introduced Bidirectional Recurrent Neural Network
	Hochreiter & Schmidhuber	introduced LSTM, solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	introduced Deep Belief Networks, also introduced layer-wise pretraining technique, opened current deep learning era.
2009	Salakhutdinov & Hinton	introduced Deep Boltzmann Machines
2012	Geoffrey Hinton	introduced Dropout, an efficient way of training neural networks

2.2.1 Supervised Learning

As Supervised Learning we always know what the target is. For example, imagine that we have a dataset from the leaves key points from 3 different plants with given the name of the plant and we are training the machine learning model based on this dataset. Now we have a trained model and, in the future, if we need to find from which plant is this leave, applying this model we are going to have the result. This type of supervised machine learning is Classification and below there is more detail on this. For the plants in the image below we are using sepal length,sepal widt,petal length,petal width as shown in Figure 1. [3]

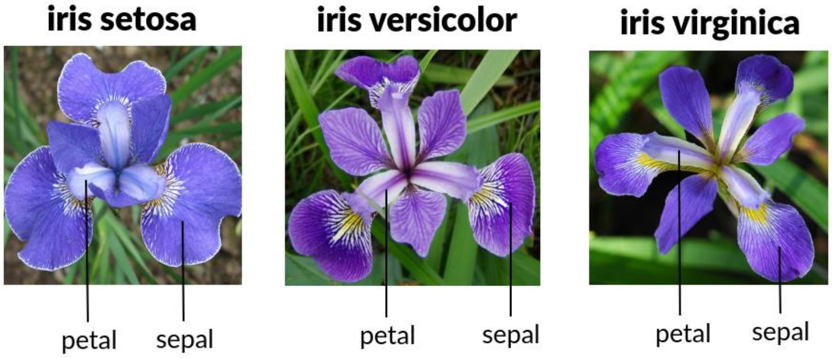


Figure 1:Iris sepals' widths and heights

2.2.1.1 Classification

As mentioned before, the classification is used to predict the category of data. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). In the Figure 2 we can see a simple 2 class problem. [4]

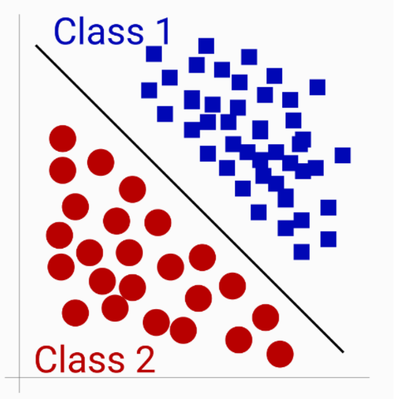


Figure 2Example of Classification

Other real-world problems that are using Classification are Credit Score, Customer Segmentation, email spam classification.

2.2.1.2 Regression

Regression is just like classification except that the response variable is continuous. In the previous example the target was to identify which plant it is, but if we need to predict the price of a house then we need to apply regression. From the mathematics we know the expression of $y = ax + b$. Imagine that y is the number of eggs that we need to have every day and x are the chickens, is familiar known that for each egg we need a female chicken and a male chicken, where the male chicken can be used for more chickens, so our equation will be $y = x + 1$. In the Figure 3 we can see exactly how this equation visualized. [5]

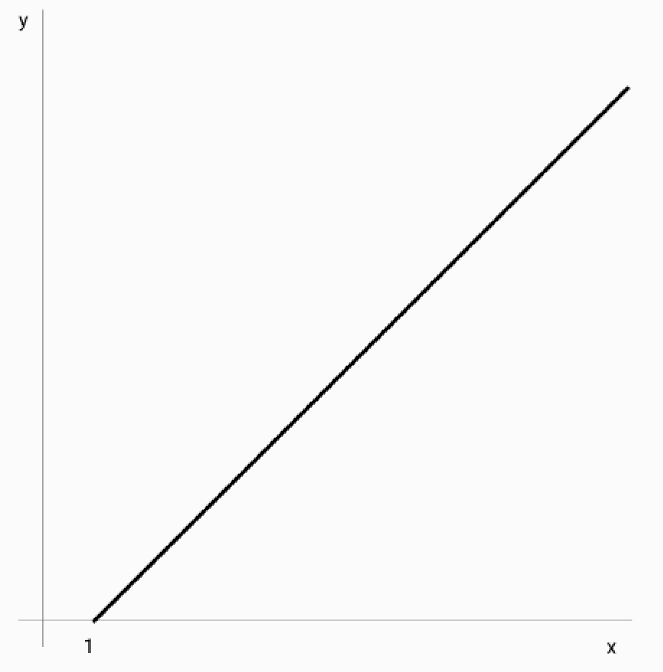


Figure 3Example of Linear Regression

2.2.2 Unsupervised Learning

We now consider unsupervised learning, where we are just given a dataset without any target and without labels on this data. To make use of this dataset, we can describe as an unsupervised machine learning model the segmentation that can help us to discover new insights from our data. For example, in how many categories can I split my customers or how many different drivers' behavior there is. In the figure let us make the hypothesis that one axis is the total amount and in the other axis the average price of their products. From the output we can realize and split our customers in 5 different segments based on their behavior. For example,in the Figure

4 the yellow refers these who spend a lot of money in cheap products while the green are these who buy a lot of expensive products. [6]

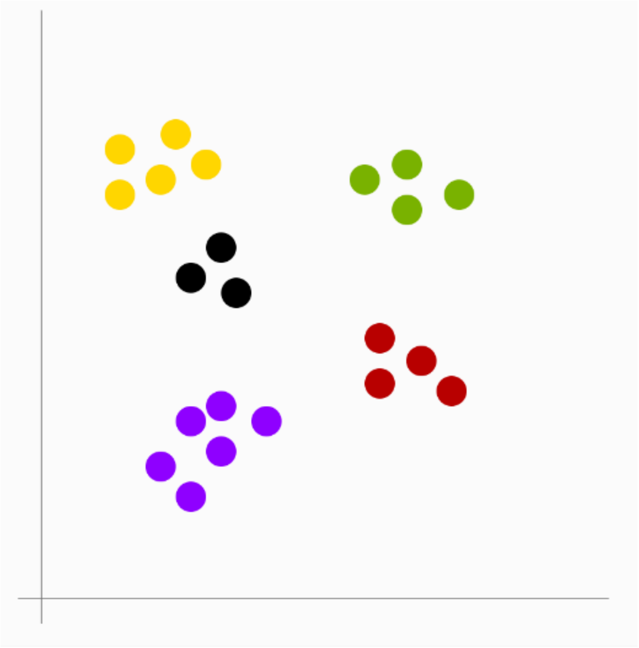


Figure 4:Clustering in Unsupervised Machine Learning

2.2.3 Reinforcement Learning

The simplest way to describe reinforcement learning is to think about having a video game and trying to improve the hard levels of the game. Then some players are going to run this game and while this is happening the model is trying to make it harder. [7]

2.2.4 Summary

To summarize in this section there are presented all the types of machine learning which are Supervised Learning, Unsupervised Learning, Reinforcement Learning.

2.3 Deep Learning

Deep learning is inspired from human’s neutral network. Neural networks are a computational model that shares some properties with the ani- mal brain in which many simple units are working in parallel with no centralized control unit. The weights between the units are the primary means of long-term information storage in neural networks. Updating the weights is the primary way the neural network learns new information. The behavior of neural networks is shaped by its network architecture. A network's architecture can be defined (in part) by the following:

- Number of neurons
- Number of layers
- Types of connections between layers

Deep learning architectures can be defined in four major architectures:

- Unsupervised Pretrained Networks
- Convolutional Neural Networks
- Recurrent Neural Networks
- Recursive Neural Networks

More major architectures are released in year. There is continuous research in deep learning and in the domain of neural networks. One of the most well-known new architecture is the Long Short Term Memory which is used from time series. In the next sectors there is a brief for each of these but before this we can look at the components of each Deep Learning Model. [8] In the Figure 5 we can see how we can represent a neural network.

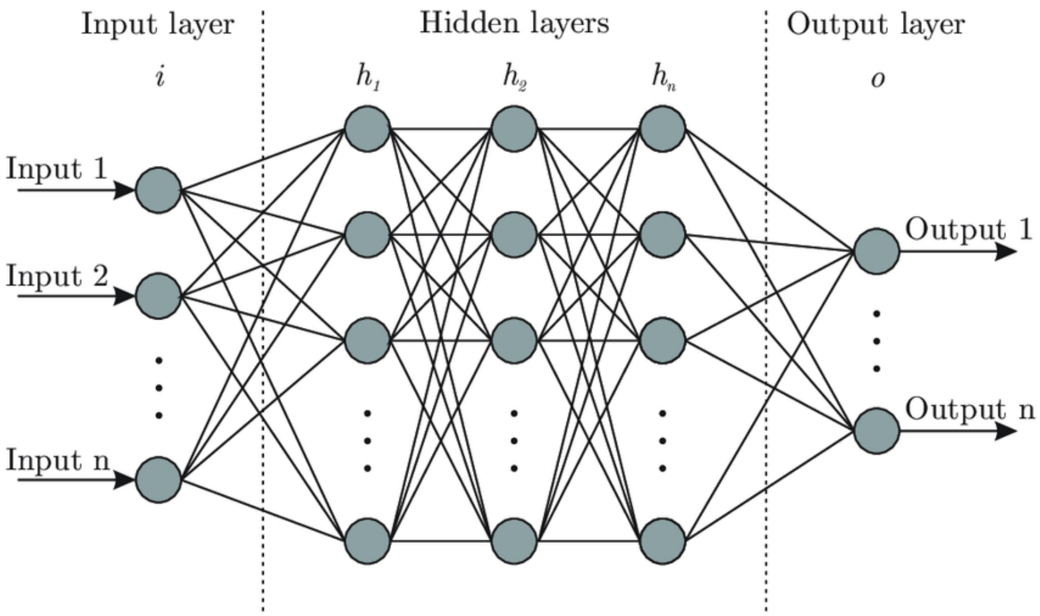


Figure 5: Neural Network

2.3.1 Components of Deep Networks

To understand how a deep network works we need to look at its components of this. Below there are the core components:

- Parameters
- Layers
- Activation functions
- Loss functions
- Optimization methods
- Hyper Parameters

2.3.1.1 Parameters

Parameters in neural networks relates with the weights on the connections while also are The coefficients in a linear regression or logistic regression. [8]

2.3.1.2 Layers

The layers are input, hidden and output layers. The Input layer is how we get input data (vectors) fed into our network. For the hidden layers, there are one or more hidden layers in a feed-forward neural network. The weight values on the connections between the layers are how neural networks encode the learned information extracted from the raw training data. [8]

2.3.1.3 Activation functions

With the Activation functions the algorithm can perform complex computation in the hidden layers and the transfers the output to the output layers. The most known activation function is Rectified linear unit (ReLU). The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time. [8]

2.3.1.4 Loss functions

The description of Loss functions is simple. For example, we calculate a metric based on the error we observe in the outputs of the model. One metric of Loss functions for the regressions is the Mean Squared Error (MSE).

2.3.1.5 Optimization methods

The optimization methods are algorithms that have been developed in order to minimize loss. Some of the methods where are most popular are:

- Stochastic gradient descent (SGD)
- Limited-memory BFGS
- Hessian-free

2.3.1.6 Hyperparameters

As hyperparameter we can define any parameter that might affect usage such as layer size. We can change the number of input layers in order to matchup more features. The TensorFlow [9] package has dropout as hyperparameter, with the dropout when we are training the model and the accuracy after many iterations is getting worse is stopping the procedure of training. [8]

2.3.2 Unsupervised Pretrained Networks

We can split this category in three segments:

- Autoencoders
- Deep Belief Networks (DBNs)
- Generative Networks (GANs)

2.3.2.1 Autoencoders

Autoencoders are a variant of feed-forward neural networks that have an extra bias for calculating the error of reconstructing the original input. After training, autoencoders are then used as a normal feed-forward neural network for activations. This is an unsupervised form of feature extraction because the neural network uses only the original input for learning weights rather than backpropagation, which has labels. Deep networks can use autoencoders as building blocks for larger networks. [8]

2.3.2.2 Deep Belief Networks

In this the invisible layer of each sub-network is the visible layer of the next. The hidden or invisible layers are not connected to each other and are conditionally independent. Deep Belief Networks in the most recent years are being replaced from Convolutional Neural Networks. [8]

2.3.2.3 Generative Networks

GANs are an example of a network that uses unsupervised learning to train two models in parallel. A key aspect of GANs (and generative models in general) is how they use a parameter count that is significantly smaller than normal with respect to the amount of data on which we’re training the network. The network is forced to efficiently represent the training data, making it more effective at generating data similar to the training data. [8]

2.3.3 Convolutional Neural Networks

Convolutional Neural Networks are being used for the processing of images. These networks can read all properties from a photograph, that means the 3 channels of RGB information. Two more things that it can get are the image width and image height. [8] In the Figure 6 we see an example of a Convolutional Neural Network and how is reading the image in order to classify the image.

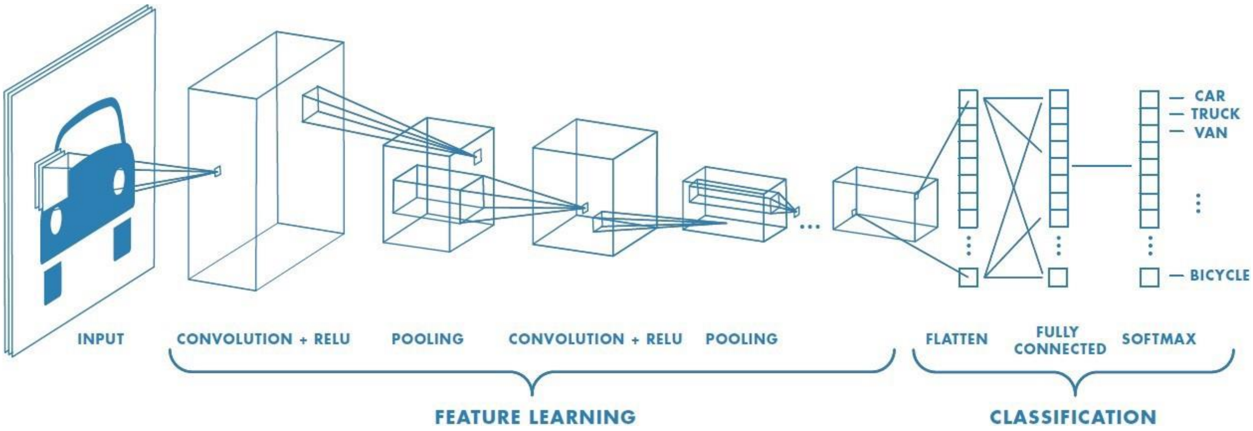


Figure 6:Convolutional Neural Network

2.3.4 Recurrent Neural Networks

Recurrent Neural Networks are a superset of feed-forward neural networks, but they add the concept of recurrent connections. These connections span adjacent time-steps, giving the model the concept of time. The conventional connections do not contain cycles in recurrent neural networks. However, recurrent connections can form cycles including connections back to the original neurons themselves at future time-steps. [8]

One of the commonly neural networks close to Recurrent Neural Network are the Long Short Term Memory. The reason of that are the better update equations and better backpropagation. From the Journal Article ‘A review on the long short-term memory model’ we have the Figure 7 which is represent the LSTM Architecture of a typical vanilla LSTM block. [10] As Greg Van Houdt, Carlos Mosquera and Gonzalo Nápoles are reporting in this Citation the reason for the success of this recurrent network lies in its ability to handle the exploding/vanishing gradient problem, which stands as a difficult issue to be circumvented when training recurrent or very deep neural networks.

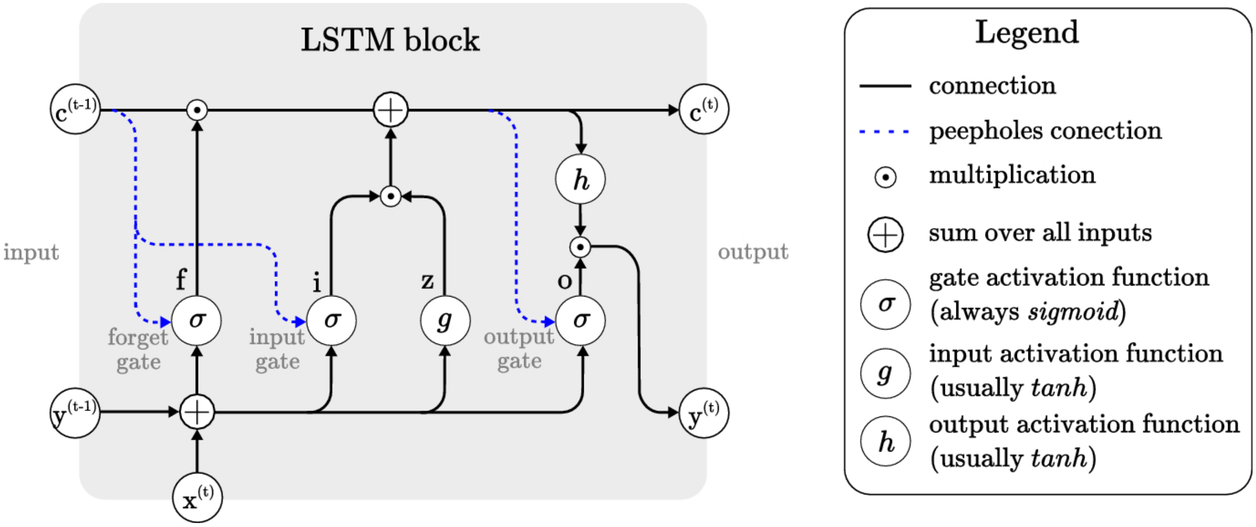


Figure 7:Architecture of a typical vanilla LSTM block

2.3.5 Recursive Neural Network

Recursive Neural Network are close with the Recurrent Neural Networks with the primary difference to be the ability to model the hierarchical structures in the training dataset. [8]

2.4 Summary

To sum up, deep learning and neural networks are a subfield o machine learning, inspired from the brain of animals in order to create models with more parameters and better accuracy that the simple machine learning models.

3. Related Work

In this section we are going to see parts of Related work, the first will be about public datasets and the approach that has been used and in the second popular trained models that can estimate the points of the body.

3.1 Related Work on Datasets

Human Pose Recognition and Human Activity Recognition is one of the most upscaling and interesting projects in the Deep Learning Models, or we can call it Computer Vision. In this moment there are many open datasets only for general use. Some of these are:

- HumanEva Dataset [11]
- Human3.6M [12]
- MPII Human Pose Dataset [13]
- PoseTrack [14]
- COCO Keypoint Detection [15]
- Leeds Sports Pose Extended [16]

Although for our scope there is no such as a public video dataset that have been made in order to create an algorithm for making right workout. Although below is some information in these implementations.

For the HumanEva Dataset Leonid Sigal, Alexandru O. Balan ,J. Black in the HUMANEVA: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion [17] they have used synchronized video from many angles in order to create an algorithm that can detect the human pose, in addition to providing performance against which future advances on this data can be measured, is designed to serve as a test-bed for future experiments with likelihood functions, prior models and inference methods within the context of Bayesian filtering. Graham W. Taylor, Leonid Sigal, David J. Fleet and Geoffrey E. Hinton in Dynamical Binary Latent Variable Models for 3D Human Pose Tracking [18] they are trying to approach the dataset with the same as the previous authors. Another approach of Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, Yichen Wei have used a 2D model and 3D model in order to create an algorithm that can read better and classify the human pose in Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach [19], they have used for their experiments the datasets from Human3.6M and MPII Human Pose Dataset. With the dataset of Human3.6M in Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments [20] from Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu they have introduced the ability to train complex approximations to non-linear models on millions of examples opens up possibilities to develop alternative feature descriptors and correlation kernels, and to test them seamlessly, at large scale. Mykhaylo Andriluka1 Leonid Pishchulin, Peter Gehler and Bernt Schiele in 2D Human Pose Estimation: New Benchmark and State of the Art Analysis [21] they are analyzing the MPII Human Pose Dataset based on the camera position and establishing new qualitatively higher standards for evaluation and analysis of pose estimation methods. While the MPII Human Pose Dataset is one of the datasets where there have been written many citations on this, Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang in Deep High-Resolution Representation Learning for Human Pose Estimation [22] are maintaining the high resolution for the whole process and fuse multi-resolution representations repeatedly, at this citation the authors have used also the COCO Keypoint Detection for their experiments. Vasileios Belagiannis and Andrew Zisserman in Recurrent Human Pose Estimation [23] have introduced a recurrent human model for 2D human pose estimation using MatConvNet [24] in the dataset MPII Human Pose Dataset. Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, Bernt Schiele in Learning to Refine Human Pose Estimation [25] are using MPII Human Pose and PoseTrack to further improve the TensorFlow open-source framework. Feng Zhang, Xiatian Zhu, Mao Ye in Fast Human Pose Estimation [26] are trying to minimize the time of processing images in order to create a fast model. Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, Bernt Schiele in PoseTrack: A Benchmark for Human Pose Estimation and Tracking [27] they have enables objective comparison of different approaches for articulated people tracking in realistic scenes. Last one but interested citation is from Ijaz Akhter, Michael J. Black the Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction [28] where they have create an algorithm based on the body angles in order to have the best results which is very promising for the future as they are going to have limits in these angles.

3.2 Related Work on Open-Source Frameworks

There are many open-source frameworks in order to use as solution to detect human body key points.Some of the Frameworks are:

- DeepPose [29]
- OpenPose [30]
- Posenet by TensorFlow [31]
- 3D Human Pose Estimation by OpenVINO [32]
- Mediapipe’s Pose [33]

The DeepPose network has been developed with the usage of Deep Neural Networks and they are using ar reference point to begin the detection of the body two points, the arms and the legs in order to compile the rest points of the body. [29] On the other hand, the OpenPose has been created based on Convolutional Neural Networks and one of the advantages is that is able to detect multi humans pose. [30] Posenet by Tensorflow has been built in order to be used in smartphones, so it is really fast to use in a computer. [31] OpenVINO has also built a model with some demos and promising results. [32] The last one is Mediapipe's Pose model, this model has been inspired by the Blaze face model and Leonardo's Vitruvian man. [34]

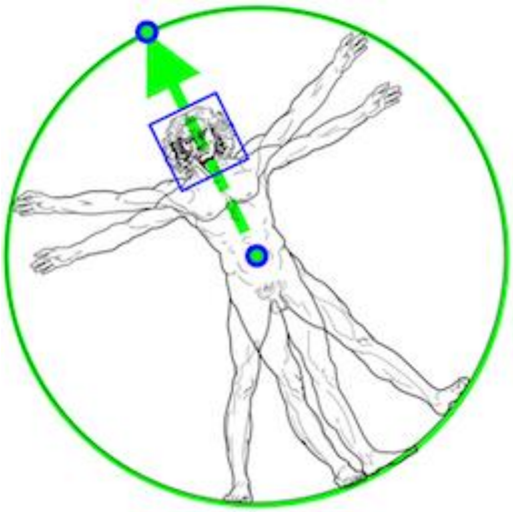


Figure 8: Vitruvian man aligned via our detector vs. face detection bounding box.

4. The implementation of our work

In this section we are going to describe how our implementation works from how and why we selected a specific framework and the results of this work.

4.1 Selected Framework

In the previous section, we have presented a simple description for the most known frameworks for human pose estimation. In the early stages of this work the initial selection was PoseNet by TensorFlow, although after a comparison with Mediapipe's Pose the last was having much smoother and faster results, while Mediapipe's is using a normalization in the data in order to achieve this smoothness. To be more in detail mediapipe's Pose is starting the calculation of body based on the face, after calculating this is trying based on the Vitruvian man to predict the middle point between the person's hips, the size of the circle circumscribing the whole person, and incline (the angle between the lines connecting the two mid-shoulder and mid-hip points). This framework output has got 33 points where in the Table 2 there are in detail the Keypoint names and in Figure 9 these points in a body.

Table 2: Mediapipe's Pose Keypoints

Keypoint	Description
0	NOSE
1	LEFT_EYEINNER
2	LEFT_EYE
3	LEFT_EYEOUTER
4	RIGHT_EYEINNER
5	RIGHT_EYE
6	RIGHT_EYEOUTER
7	LEFT_EAR
8	RIGHT_EAR
9	LEFT_MOUTH
10	RIGHT_MOUTH
11	LEFT_SHOULDER
12	RIGHT_SHOULDER
13	LEFT_ELBOW
14	RIGHT_ELBOW
15	LEFT_WRIST
16	RIGHT_WRIST
17	LEFT_PINKY
18	RIGHT_PINKY

19	LEFT_INDEX
20	RIGHT_INDEX
21	LEFT_THUMB
22	RIGHT_THUMB
23	LEFT_HIP
24	RIGHT_HIP
25	LEFT_KNEE
26	RIGHT_KNEE
27	LEFT_ANKLE
28	RIGHT_ANKLE
29	LEFT_HEEL
30	RIGHT_HEEL
31	LEFT_FOOTINDEX
32	RIGHT_FOOTINDEX

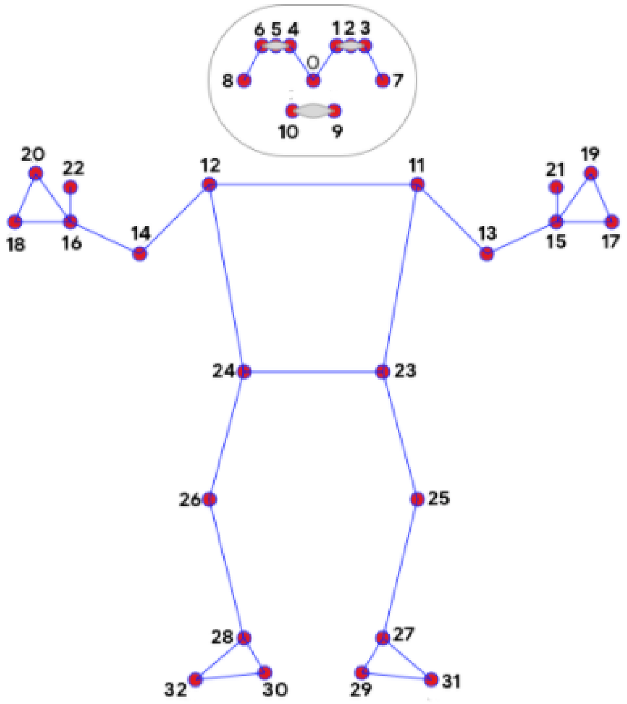


Figure 9:Mediapipe's Pose Keypoints in body

Some other features that help us to make the decision to use this framework,were the static image mode, minimum detection confidence and the pose landmarks. The static image mode will try to detect the most prominent person in the very first images, and upon a successful detection further localizes the pose landmarks. In subsequent images, it then simply tracks those landmarks without invoking another detection until it loses track, on reducing computation and latency. The minimum detection confidence helps and sets a limit in the person detection, using that can set us a limit in data quality. Last one the pose landmarks, it returns three values, x, y, z. The first two x and y are values from 0 to 1 in order to give us the position of each keypoint where 0 is the upper left point of the frame and 1 the down right. Last one is z, the z represents the landmark depth with the depth at the midpoint of hips being the origin, and the smaller the value the closer the landmark is to the camera. One more thing that we need to mention for the mediapipe's pose model is the only one with full documentation, online demos using Google Colab and Citation. [33]

4.2 Dataset

As it has been mentioned before there is no public dataset to work with. For this reason,there have been created 3 videos for 3 different exercises. These videos have been recorder in outdoor place at afternoon hours in order to have enough light for the details of the video with a digital Single Lens Reflex (DSLR camera) recorded in Full HD Quality 1920 x 1080 and 30 frames per second (fps) in mp4 format. The model of the camera was Canon DSLR EOS 2000D with lens model BK 18-55 SEE [35]. The first video is a workout for lifting dumbbells in an upright position, with 6 repetitions of the workout. The second video is the workout of out shoulder turbs with dumbbells with 8 repetitions and the last one is hand stretches with dumbbells with 7 repetitions. The person in these videos is me. In Figures 10,11,12 there are screenshots from each video.

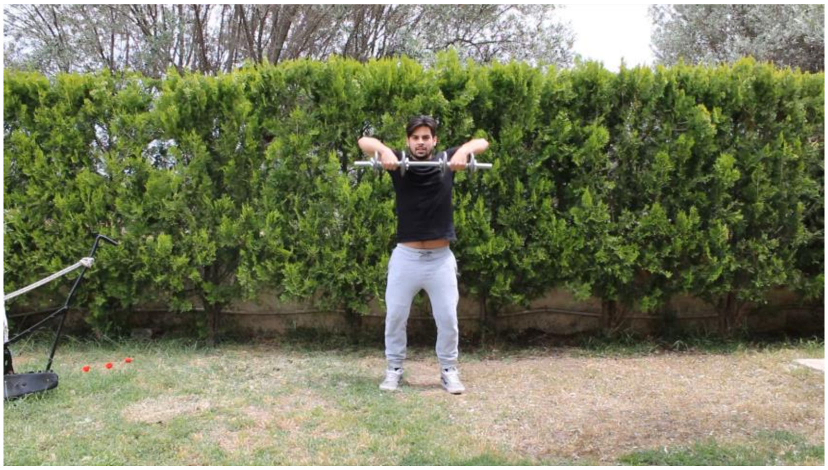


Figure 10: Lifting dumbbells in an upright position.

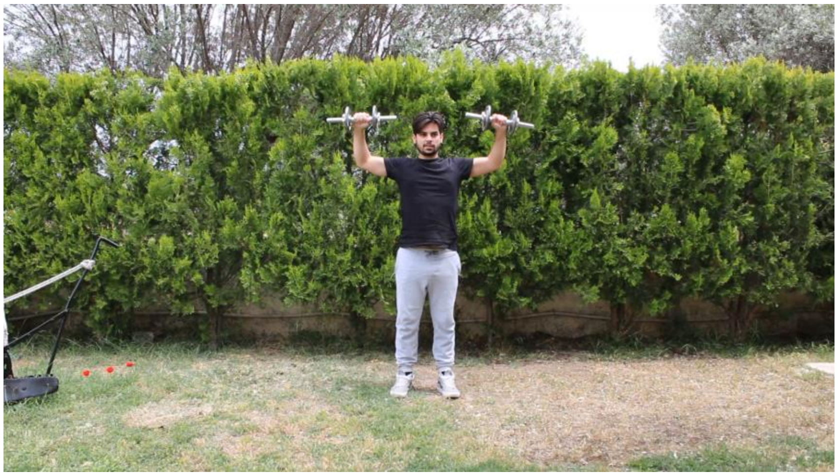


Figure 11:Out Shoulder turns with dumbbells.

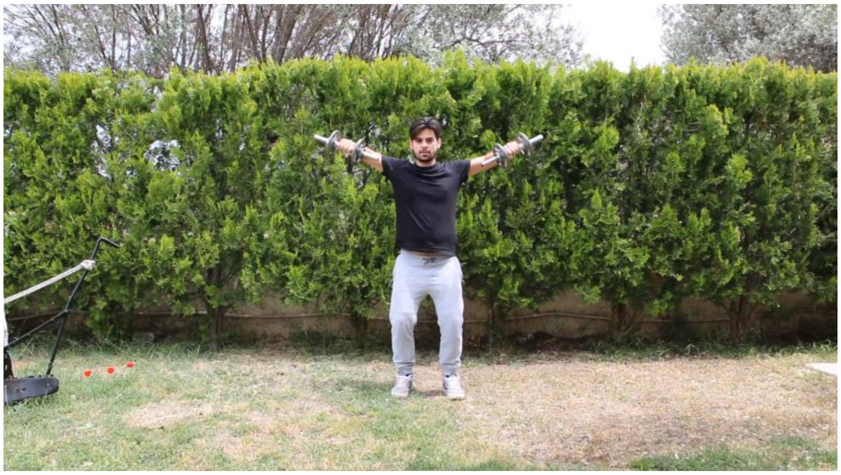


Figure 12: Hand stretches with dumbbells.

4.3 The Workouts' instructions

For the workout instructions it has been used a Greek book the "Weight Training" from Ioannis Fatouros and Thanasis Chatzinikolaou [36]. The three exercises that we have implemented are:

- Lifting dumbbells in an upright position.
- Out Shoulder turns with dumbbells.
- Hand stretches with dumbbells.

In order to create an algorithm which can detect the points of the body we have included the instructions of the book above. The algorithm does contain the follow check points:

- Symmetry between left and right side.
- Height of wrist versus the point of shoulder.

- Bent knees.
- The point of wrist versus the point of hips.
- Stable the rest points of the body.
- Wrist front of elbow.
- Wrist position versus elbow position.

In the next sections there are the algorithms for each one of the above that have been used.

4.3.1 Symmetry Between left and right side

The symmetry algorithm is the one of the most complicated, and in order to recognize the symmetry from the right to the left side it has been used 3d vector calculation between the points of wrist-hip, elbow-hip, wrist-shoulder. For example, for each pair of the above it has been used the Mathematical type:

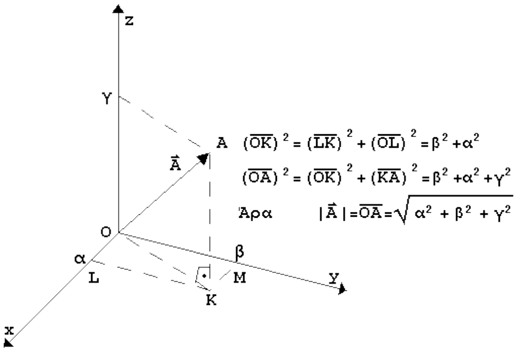

z
Y
A
$\vec{A}$
$(\overline{OK})^2 = (\overline{LK})^2 + (\overline{OL})^2 = \beta^2 + \alpha^2$
$(\overline{OA})^2 = (\overline{OK})^2 + (\overline{KA})^2 = \beta^2 + \alpha^2 + \gamma^2$
Αρα
$|\vec{A}| = \overline{OA} = \sqrt{\alpha^2 + \beta^2 + \gamma^2}$
O
β
α
L
M
Y
K
x

Figure 13:3d Vector

$$\left|\vec{A}\right| = \overline{OA} = \sqrt{\alpha^2 + b^2 + C^2}$$

After applying this above, we are aggregating left and right side of the body in order to compare the the distances for each pair. As a last step we compare these differences for each frame, if there is a difference in the frames then we assume that there is no symmetry on the body. The 3d vector has used the 3 variables from mediapipe pose x, y, z, in the Figure 14 we represent these points.

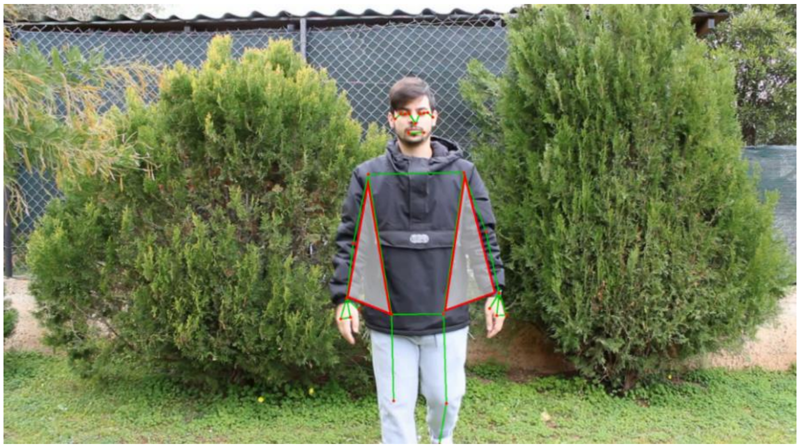


Figure 14: Symmetry Distances

4.3.2 Height of wrist versus the point of shoulder

The two exercises that have been implemented request that the height of the wrist is going to be maximized at the height of the shoulder. In order to achieve this and to have more accurate results we have applied in our data the mean moving average for 10 frames. For each point of the body, mediapipe returns the x,y,z and in this check we need that the y of wrist must be lower or equal from the y of shoulder.

4.3.3 Bent Knees

All exercises and not only these in this report, request and need the bent knees. The reason is for the human body’s waist do not be pressed and not lifting all the weights at this point of the body. The implementation for that was simple, it was just to have the z value negative. The z value as we have mentioned before when is 1 it means that is close to the camera, using this functionality from the mediapipe package we make the check that z of both knees must always be zero or negative.

4.3.4 The point of wrist versus the point of hips

One of the exercises needs the wrist and the hips from each side to be in the same straight. For that have been used the x values for each side with a mean moving average. As in the previous check, in this one we are using the x value from the mediapipe model.

4.3.5 Stable the rest points of the body

For this check there have been used nine points, the Nose, left and right hip, left and right knee, left and right Ankle. As the next step is that all these points from the beginning of the video until the end do not move more than 5% from the mean point from all the videos. This mean point is using combined all three values from the points x, y, z. The reason about applying this check is simple, while we are making any exercise the body must be stable in order to train more the specific muscles in the exercise.

4.3.6 Wrist front of elbow

To check which point is front from another it has been used the same way as in the bent knees. The z of wrist to be always higher than elbow. This is a simple check in order to assume in a 3d bases that the wrist is always in front of elbow.

4.3.7 Wrist position versus elbow position

This is only for the exercise of Out Shoulder turns with dumbbells and is similar with the approach of the point of wrist versus the point of hips. The point of wrist must be for x axis in the same straight as elbow.

4.3.8 Elbow position versus Shoulder

This check is also for one exercise of Out shoulder turns with dumbbells and the height of elbow must be in the same straight as the height of shoulder.

4.3.9 Total loops of Exercise

An addition to the code that has been completed is the loops of the workout. In order to the user to have the ability to understand how many loops have been completed the app counts them.

4.4 The app

For the implementation of all these it has been used the programming language Python [37] and the version 3.8. For data analysis and for the algorithms the packages of Pandas and NumPy [38] [39]. And for the user interface the app of Streamlit [40].

4.4.1 Pandas

The package of Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. The reason for this is because it is using a simple architecture of tables and can make any transformation that you need in the tables.

4.4.2 NumPy

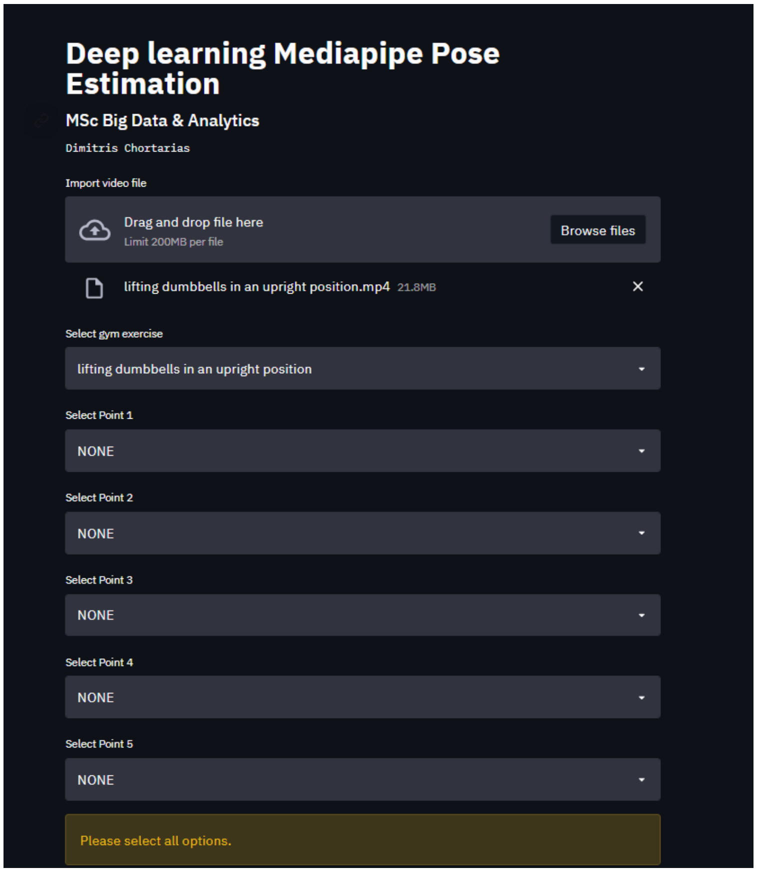
The NumPy package has used addition in pandas in order to make some transformations fasters.

4.4.3 Streamlit

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

4.4.4 The Flow of the App

The app to start working it requests at the beginning the video to be imported, the next step is to select the workout exercise and 5 points of the body like we described them before. The video must be at least in Full HD (1080p) quality and 30fps in order to have enough quality for processing. The position of the camera must be in front of the person who performs the workouts and at the same height as the eyes. Before importing the video in the app must be cropped from the moment where the workout begins. At Figure 15 we can see these prerequisites.



Deep learning Mediapipe Pose Estimation

MSc Big Data & Analytics

Dimitris Chortarias

Import video file

lifting dumbbells in an upright position.mp4 21.8MB

Select gym exercise: lifting dumbbells in an upright position

Select Point 1: NONE

Select Point 2: NONE

Select Point 3: NONE

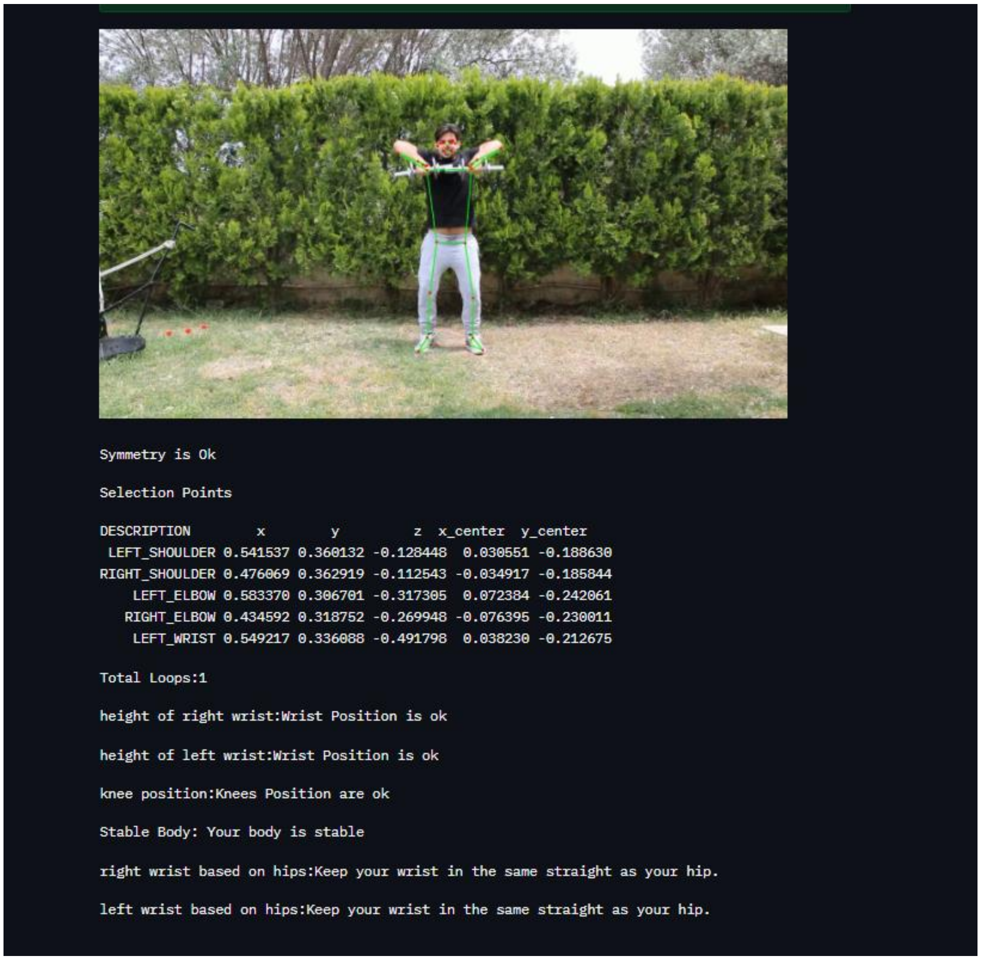
Select Point 4: NONE

Select Point 5: NONE

Please select all options.

Figure 15: The prerequisites of the app

After the completion of the fields above, the app is using the Mediapipe's Pose model in order to create the results of the points and the app is writing all these to a simple dataframe in pandas. The procedure of this model can takes for 15 seconds video almost 20-25 seconds which depends on the hardware of the pc. In this procedure in order to have not a lag in the output video the app is downscaling the video to a standard video format. The output is controlled frame by frame in order to print the results of the checks for each selected exercise. In Figure 16 there is a print screen of the outputs in a specific frame.



```
Symmetry is Ok

Selection Points

DESCRIPTION        x        y         z  x_center  y_center
 LEFT_SHOULDER 0.541537 0.360132 -0.128448  0.030551 -0.188630
RIGHT_SHOULDER 0.476069 0.362919 -0.112543 -0.034917 -0.185844
    LEFT_ELBOW 0.583370 0.306701 -0.317305  0.072384 -0.242061
   RIGHT_ELBOW 0.434592 0.318752 -0.269948 -0.076395 -0.230011
    LEFT_WRIST 0.549217 0.336088 -0.491798  0.038230 -0.212675

Total Loops:1

height of right wrist:Wrist Position is ok

height of left wrist:Wrist Position is ok

knee position:Knees Position are ok

Stable Body: Your body is stable

right wrist based on hips:Keep your wrist in the same straight as your hip.

left wrist based on hips:Keep your wrist in the same straight as your hip.
```

Figure 16: Printing results on the playback

After the playback of the Video the app is printing the aggregated results of the checks and graphs for the points of Shoulder, Elbow, Wrist, Hip per frame. In the Figure 17 is the flow of the app and how it works in order to proceed to a result.

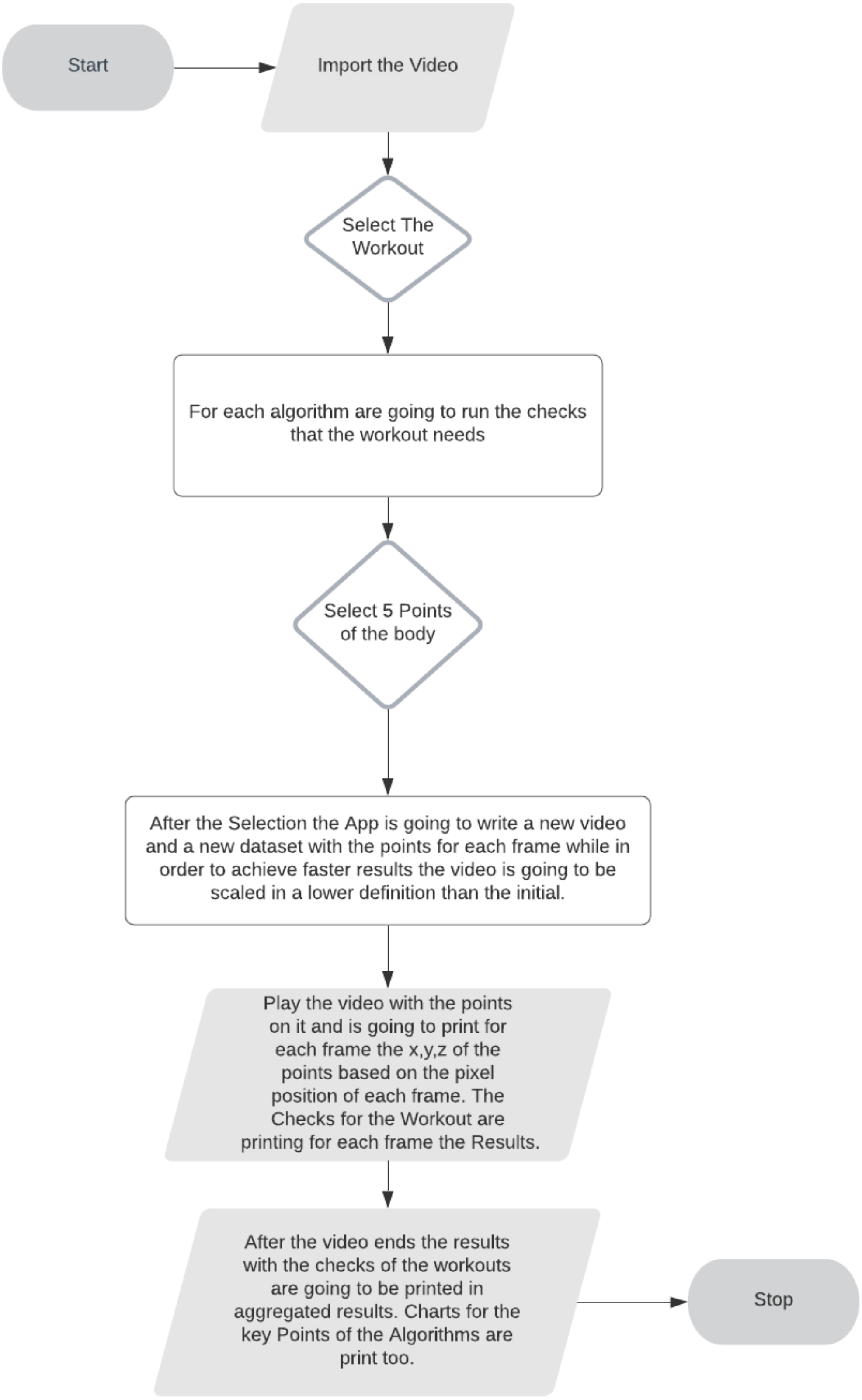

Start
Import the Video
Select The Workout
For each algorithm are going to run the checks that the workout needs
Select 5 Points of the body
After the Selection the App is going to write a new video and a new dataset with the points for each frame while in order to achieve faster results the video is going to be scaled in a lower definition than the initial.
Play the video with the points on it and is going to print for each frame the x,y,z of the points based on the pixel position of each frame. The Checks for the Workout are printing for each frame the Results.
After the video ends the results with the checks of the workouts are going to be printed in aggregated results. Charts for the key Points of the Algorithms are print too.
Stop

Figure 17: App Flow

4.4.5 Instructions to run the application

To run the app must followed the steps below:

- From the folder where the python files are saved run ‘Streamlit run main.py’
- At the chrome window which appears upload the video with the workout
- Select the video’s workout
- Select 5 points of the body which you want to have the points for each frame.

After implementing the 5 previous steps the application is going to implement all the process like it is in Figure 17. At the third step the application decides which functions should use.

Lifting dumbbells in an upright position:

- Bent Knees
- Symmetry between left and right side
- Stable the rest body
- Height of wrist versus the point of shoulders
- The points of wrist versus the point of shoulders

Out Shoulder turns with dumbbells:

- Bent Knees
- Symmetry between left and right side
- Stable the rest body
- Elbow position versus shoulder
- Wrist position versus elbow position

Hand stretches with dumbbells.

- Bent Knees
- Symmetry between left and right side
- Stable the rest body

5. Conclusion and future work

5.1 Results

Running all the above algorithms and functions the code produce the results in Table 3. The accuracy in the table below is calculated by the number of frames where the workout is executed correctly divided by the total number of frames.

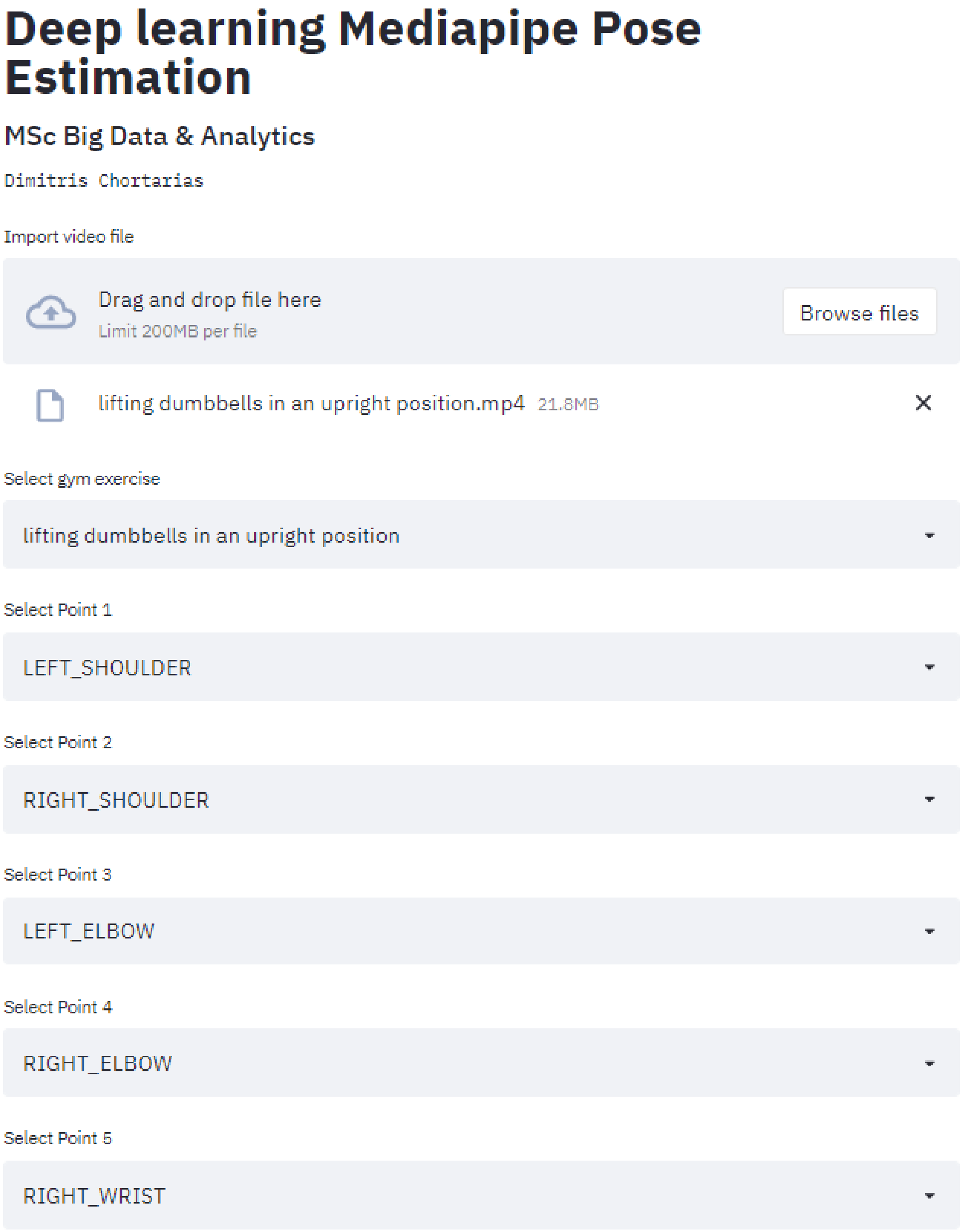
Table 3: Workout Accuracy Results

workout	*Accuracy*
Lifting Dumbbells in an upright Position	11.4%
Out Shoulder Turns with Dumbbells	0%
Hand Stretches Turns with Dumbbells	86.6%

For the First Workout of Lifting Dumbbells in an upright Position we can see at the video, that the size of dumbbells cannot help with the workout, the wrists are not in the right position based on the hips and this leads to a wrong workout with this checks the accuracy is 11.4%. The second one, Out Shoulder Turns with Dumbbells from the mediapipe pose model appears that elbows are all the time in the same straight with the shoulders or behind, the specific workout is clear at the instructions that elbows must be in front in order to not injure the shoulders, and because of this the accuracy is 0%. The last workout is one with the more Hand Stretches Turns with Dumbbells due to the level of difficulty and that is more “free” workout the algorithm shows that most of the time the accuracy is correct with 86.6%. The frames where the last workout is not ok is the symmetry between left and right side. The videos can be watched in these 3 links:

- Lifting Dumbbells in an upright Position: https://youtu.be/LAZwK6gqL2A
- Out Shoulder Turns with Dumbbells: https://youtu.be/fXlxGuHO6GA
- Hand Stretches Turns with Dumbbells: https://youtu.be/48YH1-SJi9c

5.1.1 Lifting Dumbbells in an upright Position



Deep learning Mediapipe Pose Estimation

MSc Big Data & Analytics

Dimitris Chortarias

Import video file

lifting dumbbells in an upright position.mp4 21.8MB

Select gym exercise: lifting dumbbells in an upright position

Select Point 1: LEFT_SHOULDER

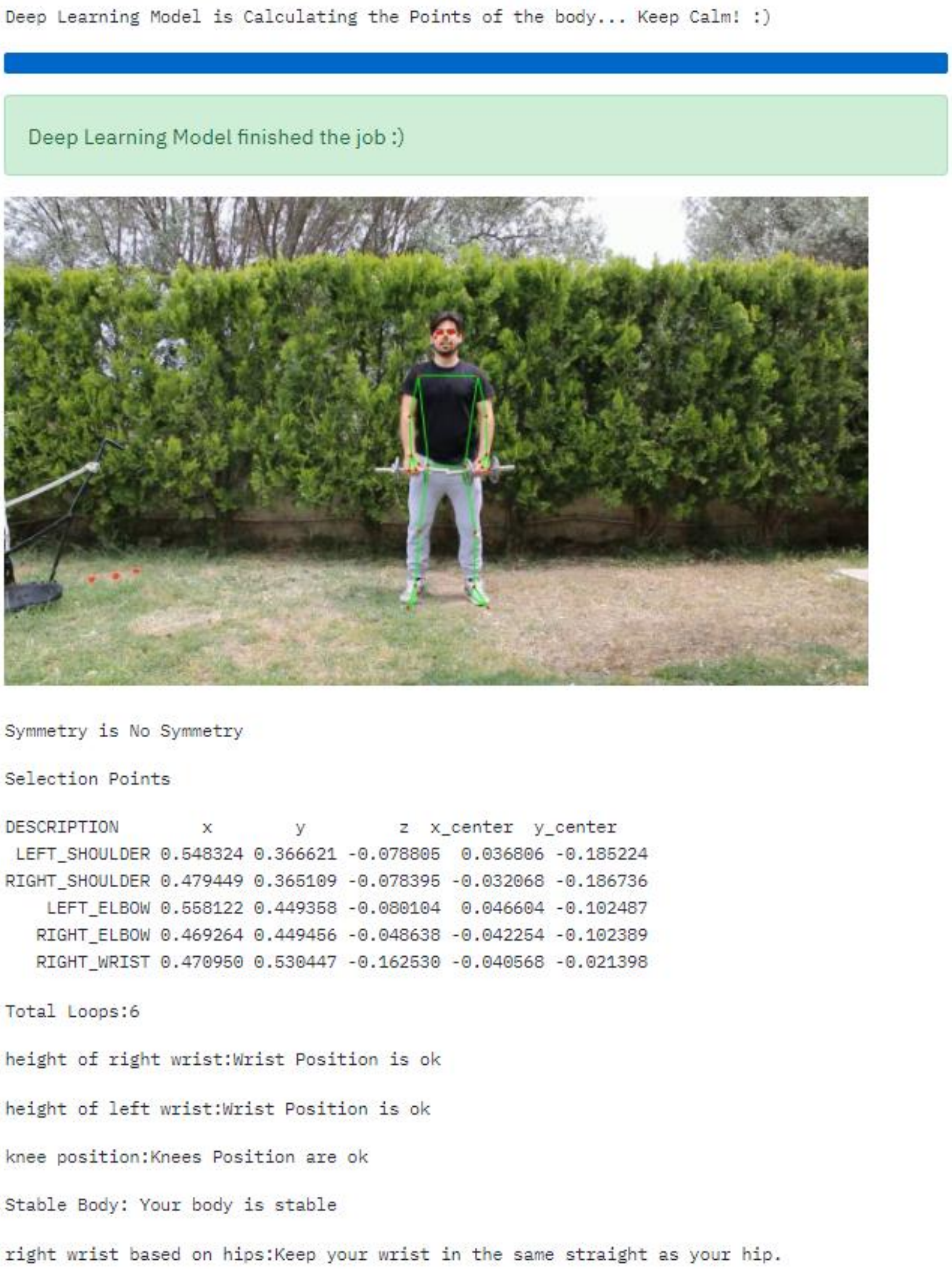
Select Point 2: RIGHT_SHOULDER

Select Point 3: LEFT_ELBOW

Select Point 4: RIGHT_ELBOW

Select Point 5: RIGHT_WRIST

Figure 18:Lifting Dumbbells in an Upright Position Part 1

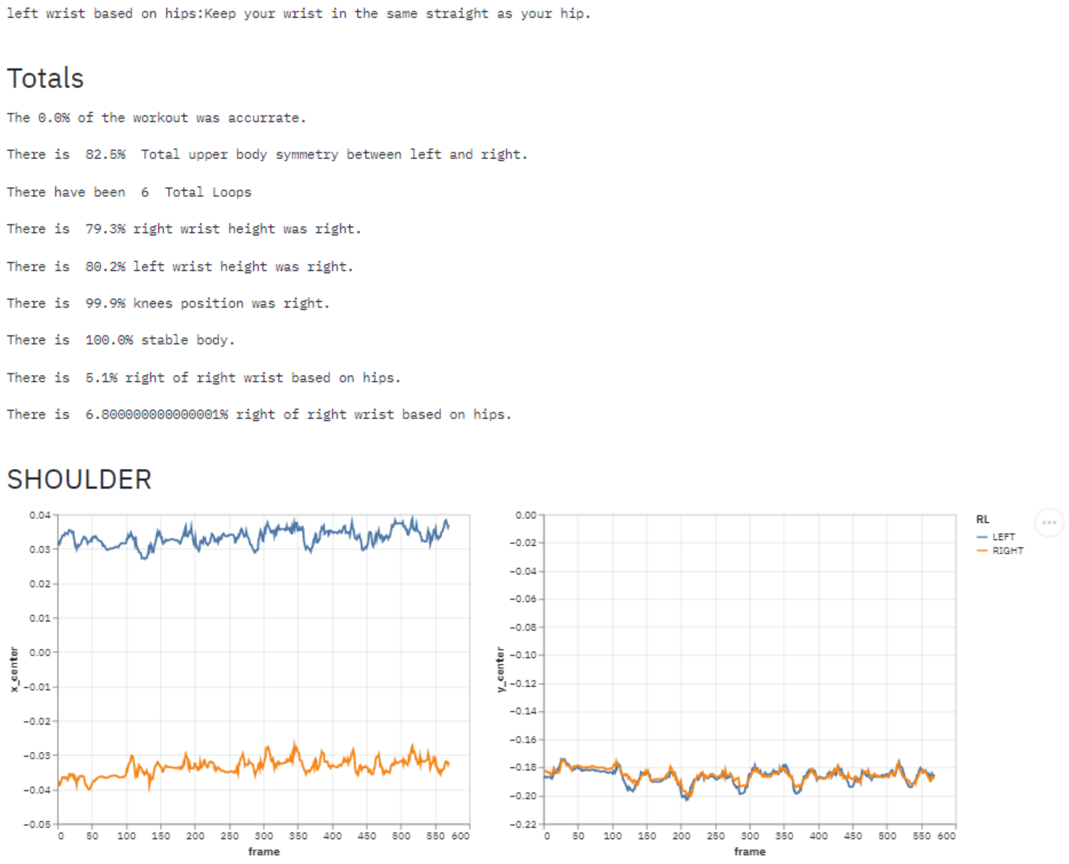


```
Deep Learning Model is Calculating the Points of the body... Keep Calm! :)
```

Deep Learning Model finished the job :)

```
Symmetry is No Symmetry

Selection Points

DESCRIPTION        x        y         z  x_center  y_center
 LEFT_SHOULDER 0.548324 0.366621 -0.078805  0.036806 -0.185224
RIGHT_SHOULDER 0.479449 0.365109 -0.078395 -0.032068 -0.186736
    LEFT_ELBOW 0.558122 0.449358 -0.080104  0.046604 -0.102487
   RIGHT_ELBOW 0.469264 0.449456 -0.048638 -0.042254 -0.102389
   RIGHT_WRIST 0.470950 0.530447 -0.162530 -0.040568 -0.021398

Total Loops:6

height of right wrist:Wrist Position is ok

height of left wrist:Wrist Position is ok

knee position:Knees Position are ok

Stable Body: Your body is stable

right wrist based on hips:Keep your wrist in the same straight as your hip.
```

Figure 19:Lifting Dumbbells in an Upright Position Part 2



```
left wrist based on hips:Keep your wrist in the same straight as your hip.
```

Totals

```
The 0.0% of the workout was accurate.

There is  82.5%  Total upper body symmetry between left and right.

There have been  6  Total Loops

There is  79.3% right wrist height was right.

There is  80.2% left wrist height was right.

There is  99.9% knees position was right.

There is  100.0% stable body.

There is  5.1% right of right wrist based on hips.

There is  6.800000000000001% right of right wrist based on hips.
```

SHOULDER

Figure 20:Lifting Dumbbells in an Upright Position Part 3

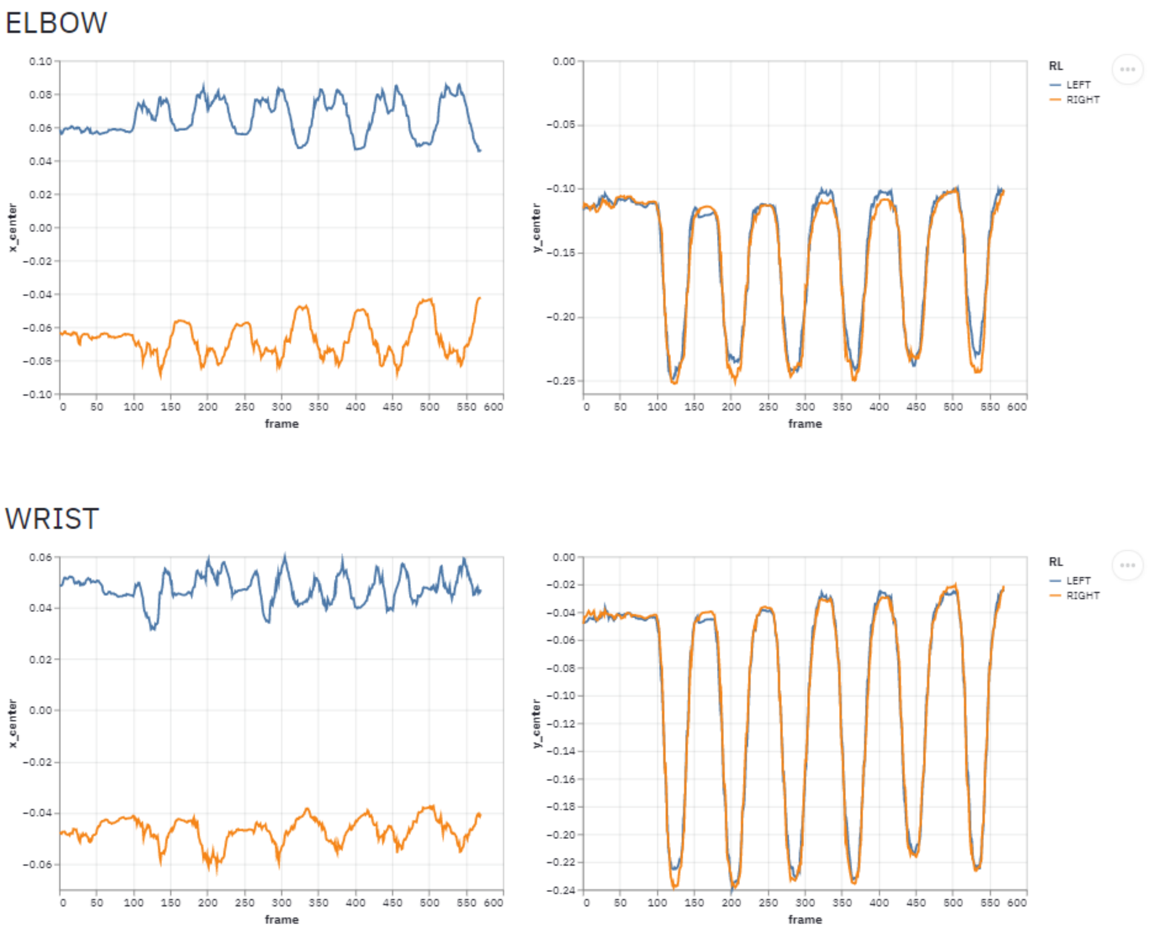


Figure 21:Lifting Dumbbells in an Upright Position Part 4

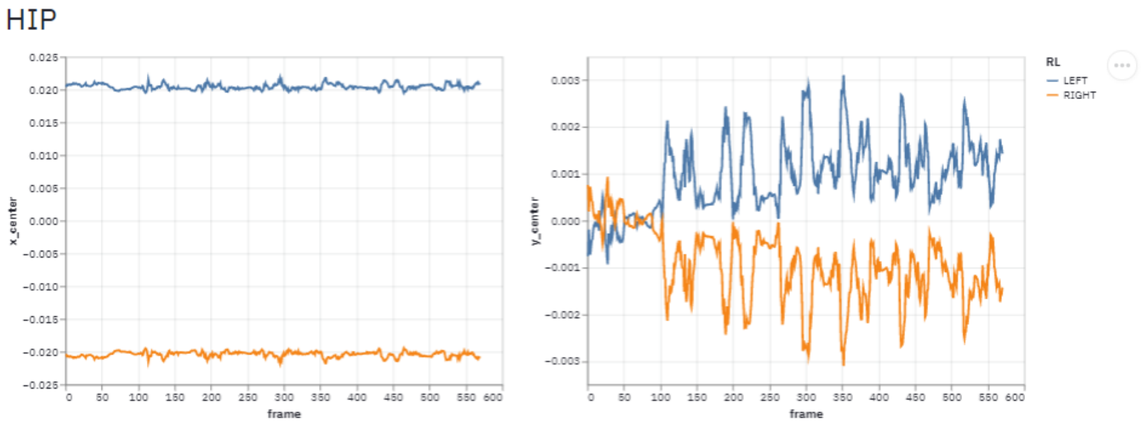
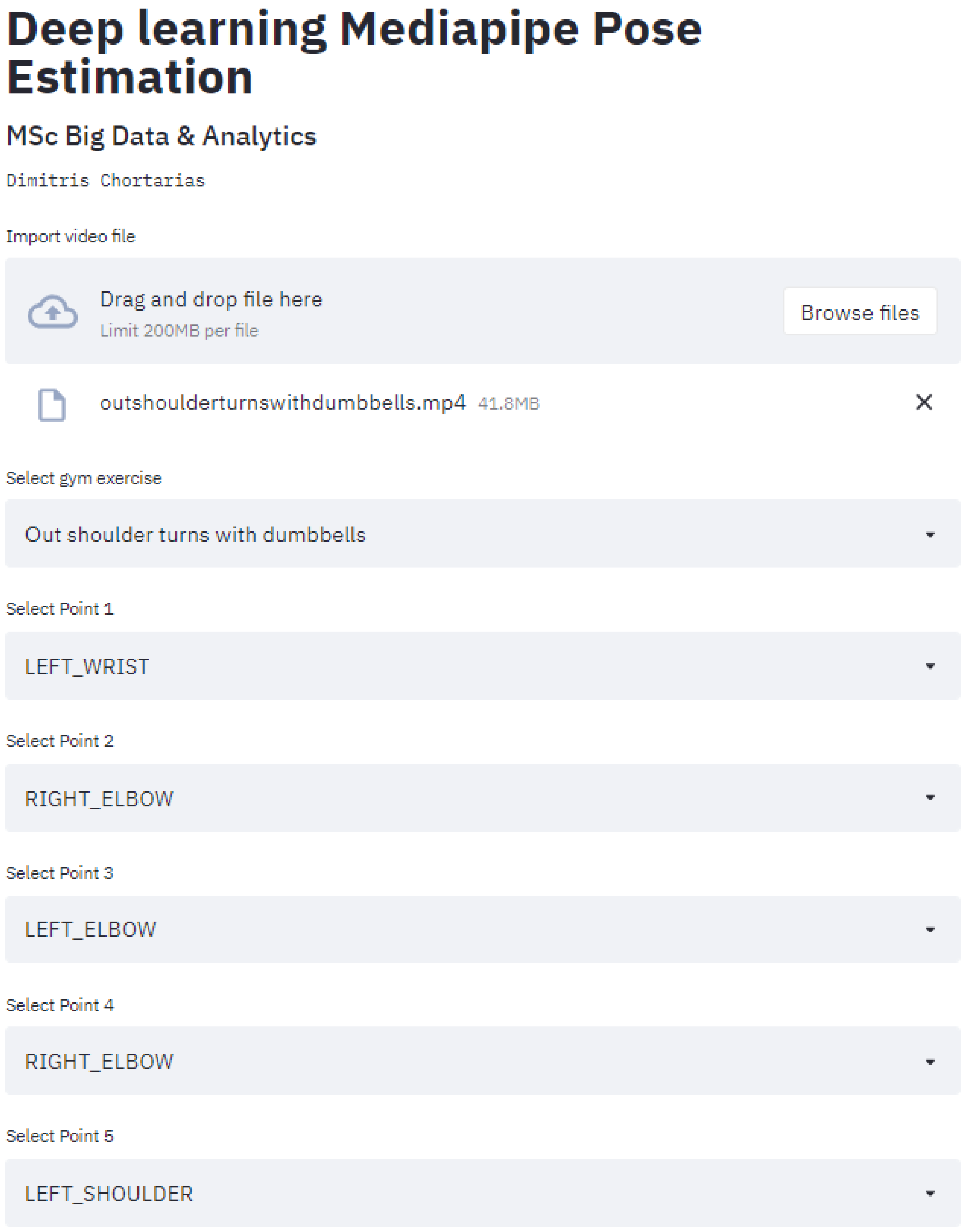


Figure 22:Lifting Dumbbells in an Upright Position Part 5

5.1.2 Out Shoulder Turns with Dumbbells



Deep learning Mediapipe Pose Estimation

MSc Big Data & Analytics

Dimitris Chortarias

Import video file

outshoulderturnswithdumbbells.mp4 41.8MB

Select gym exercise: Out shoulder turns with dumbbells

Select Point 1: LEFT_WRIST

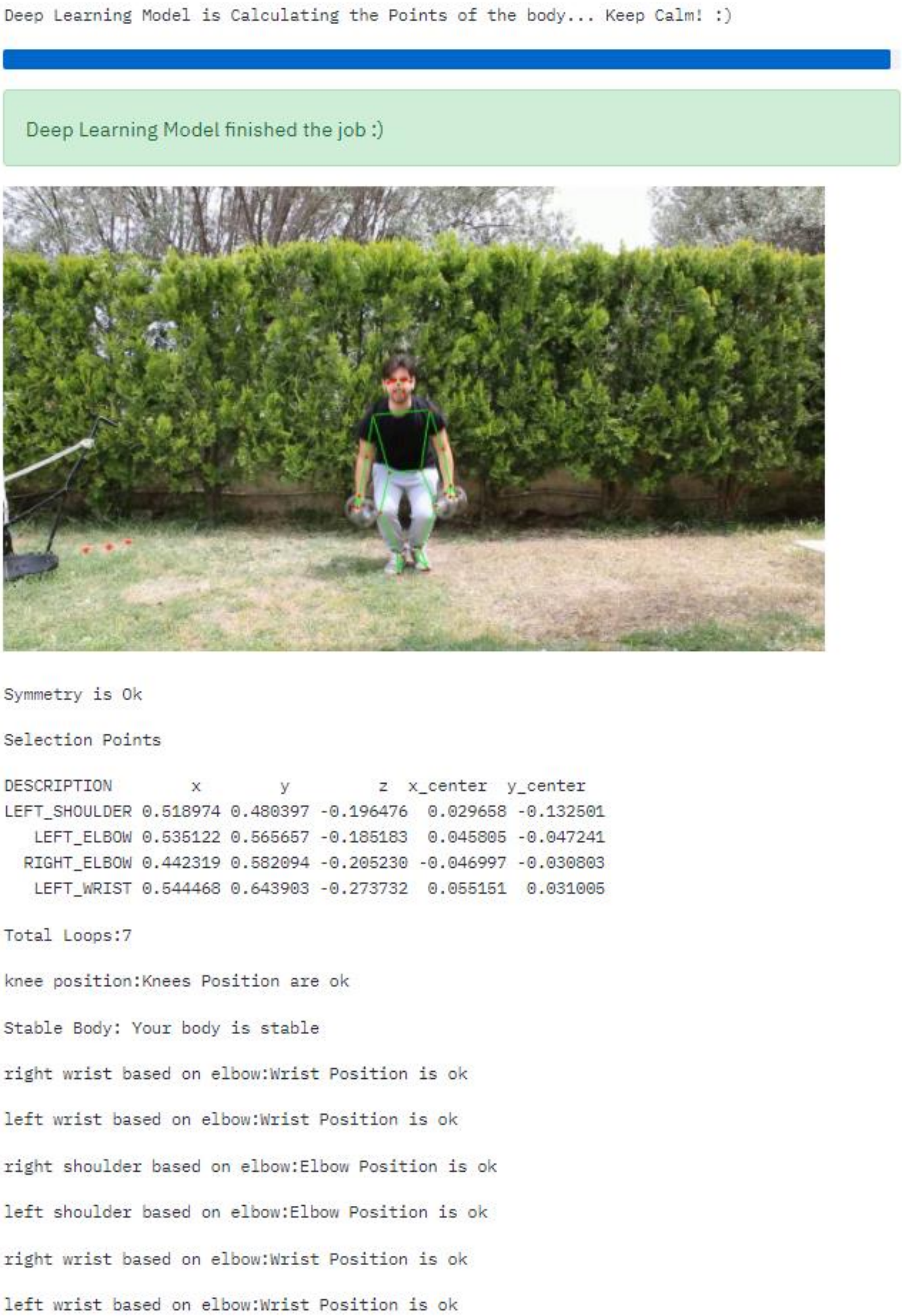
Select Point 2: RIGHT_ELBOW

Select Point 3: LEFT_ELBOW

Select Point 4: RIGHT_ELBOW

Select Point 5: LEFT_SHOULDER

Figure 23:Out Shoulder Turns with Dumbbells Part 1

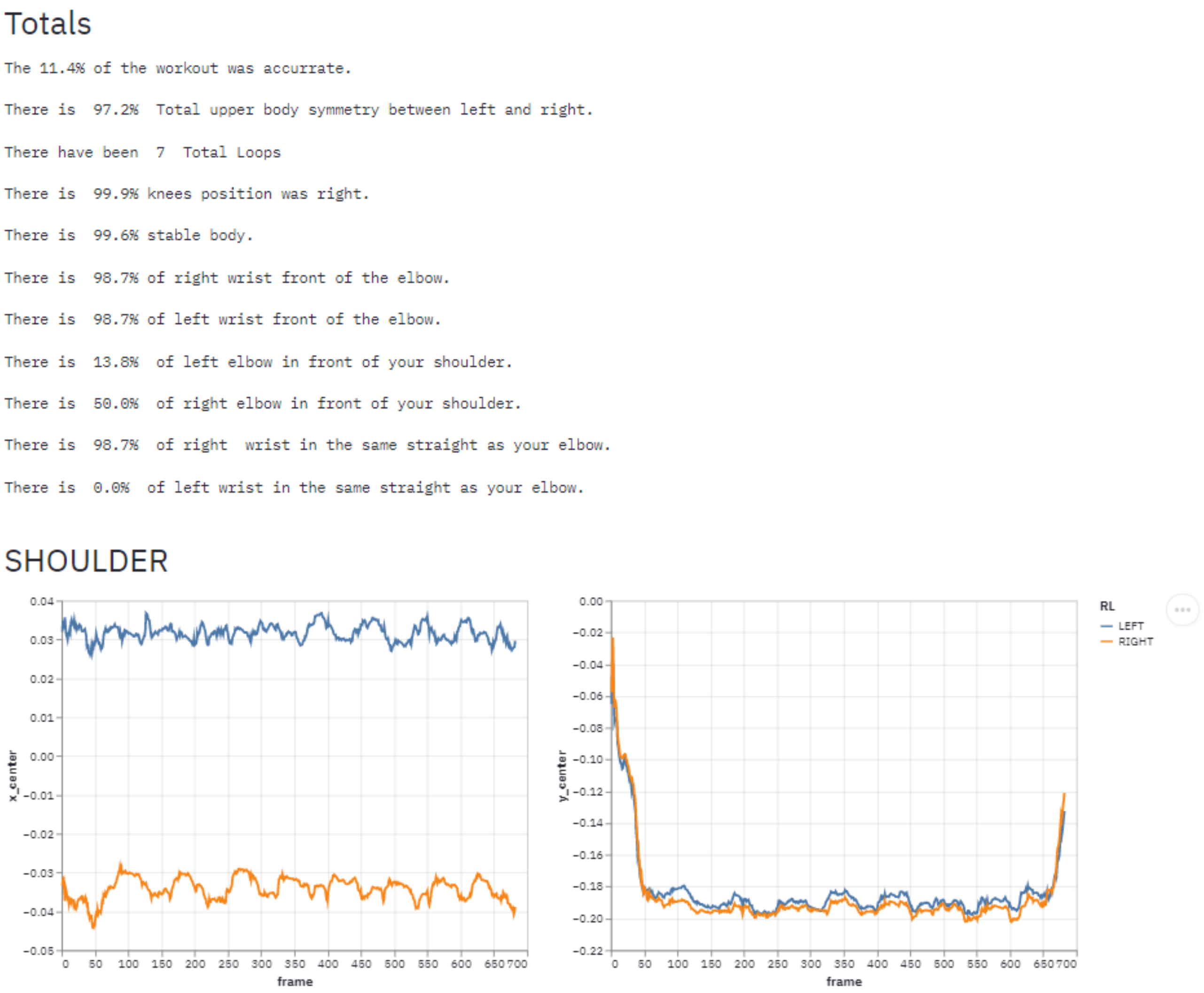


```
Deep Learning Model is Calculating the Points of the body... Keep Calm! :)
```

Deep Learning Model finished the job :)

```
Symmetry is Ok

Selection Points

DESCRIPTION        x        y         z  x_center  y_center
LEFT_SHOULDER 0.518974 0.480397 -0.196476  0.029658 -0.132501
   LEFT_ELBOW 0.535122 0.565657 -0.185183  0.045805 -0.047241
  RIGHT_ELBOW 0.442319 0.582094 -0.205230 -0.046997 -0.030803
   LEFT_WRIST 0.544468 0.643903 -0.273732  0.055151  0.031005

Total Loops:7

knee position:Knees Position are ok

Stable Body: Your body is stable

right wrist based on elbow:Wrist Position is ok

left wrist based on elbow:Wrist Position is ok

right shoulder based on elbow:Elbow Position is ok

left shoulder based on elbow:Elbow Position is ok

right wrist based on elbow:Wrist Position is ok

left wrist based on elbow:Wrist Position is ok
```

Figure 24:Out Shoulder Turns with Dumbbells Part 2



Totals

```
The 11.4% of the workout was accurrate.

There is  97.2%  Total upper body symmetry between left and right.

There have been  7  Total Loops

There is  99.9% knees position was right.

There is  99.6% stable body.

There is  98.7% of right wrist front of the elbow.

There is  98.7% of left wrist front of the elbow.

There is  13.8%  of left elbow in front of your shoulder.

There is  50.0%  of right elbow in front of your shoulder.

There is  98.7%  of right  wrist in the same straight as your elbow.

There is  0.0%  of left wrist in the same straight as your elbow.
```

SHOULDER

Figure 25:Out Shoulder Turns with Dumbbells Part 3

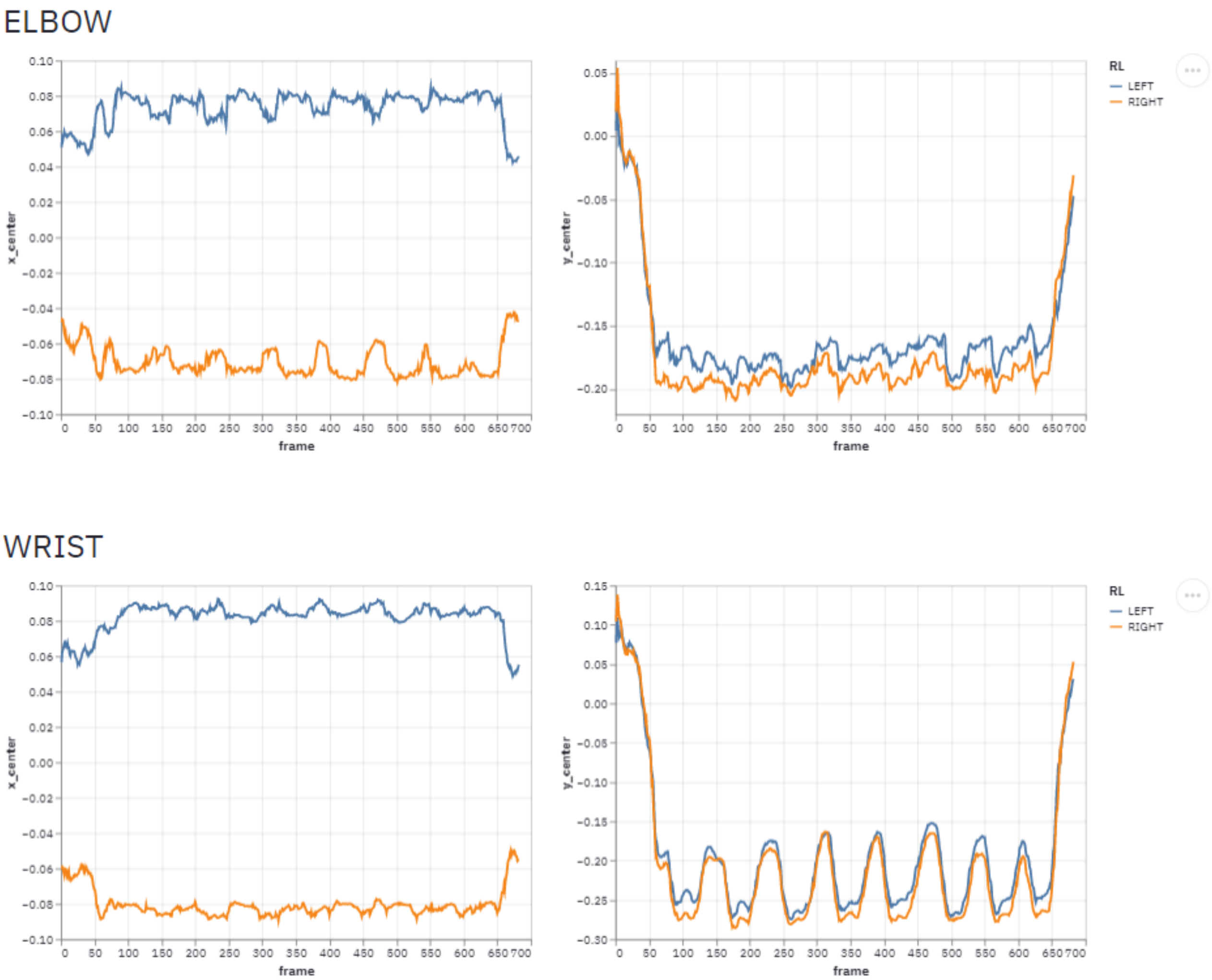


Figure 26:Out Shoulder Turns with Dumbbells Part 4

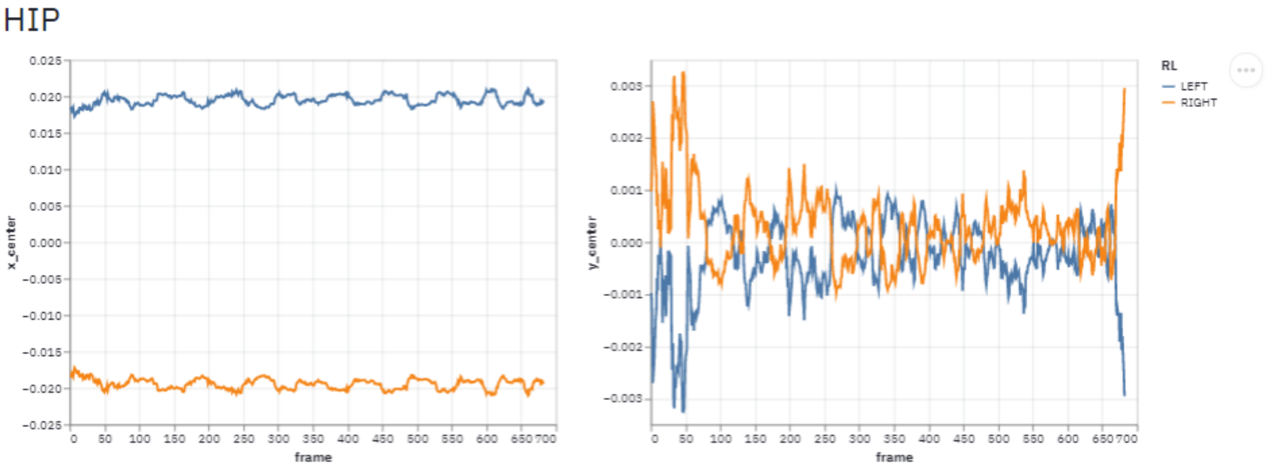
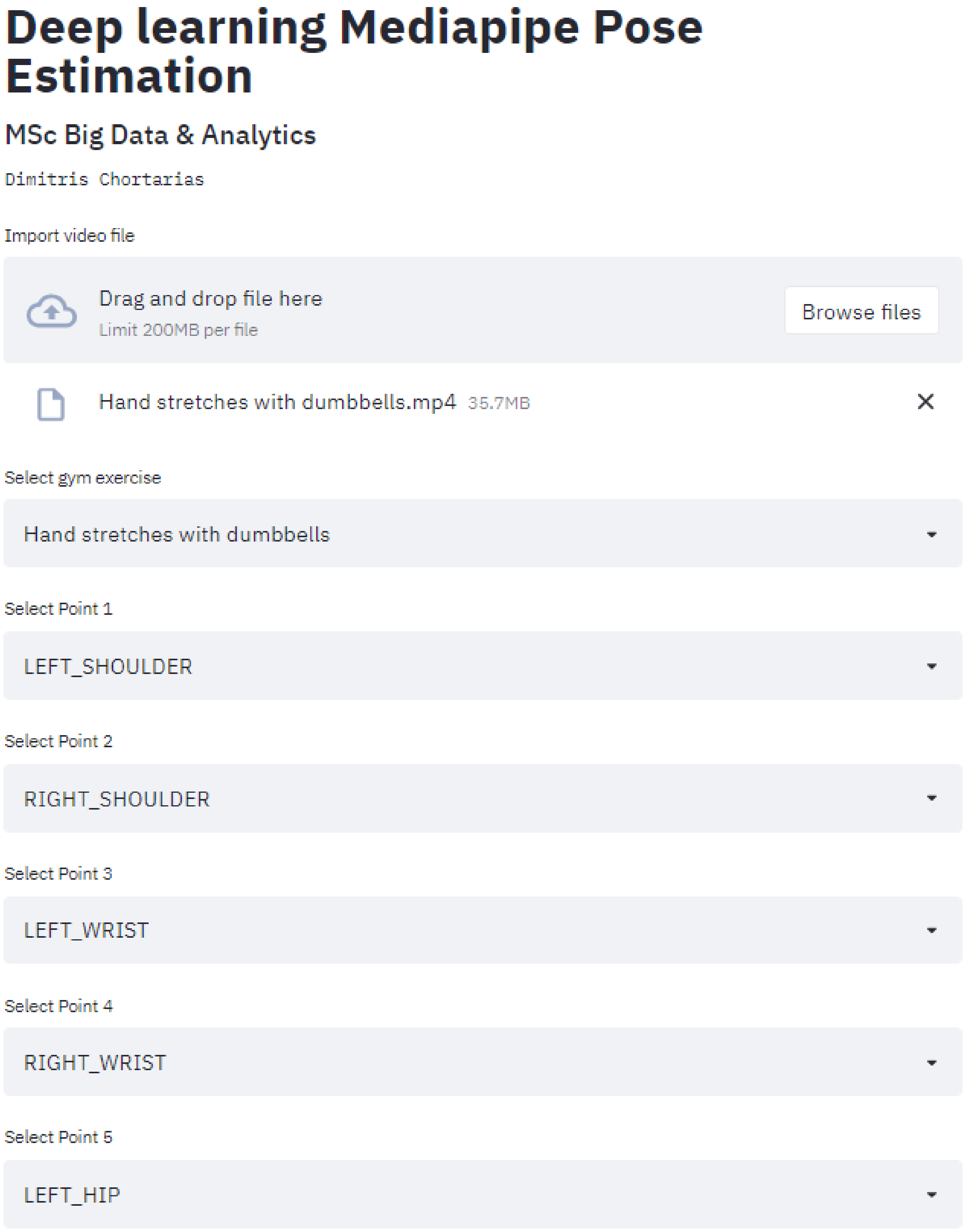


Figure 27:Out Shoulder Turns with Dumbbells Part 5

5.1.3 Hand Stretches Turns with Dumbbells



Deep learning Mediapipe Pose Estimation

MSc Big Data & Analytics

`Dimitris Chortarias`

Import video file

Hand stretches with dumbbells.mp4 35.7MB

Select gym exercise

Hand stretches with dumbbells

Select Point 1

LEFT_SHOULDER

Select Point 2

RIGHT_SHOULDER

Select Point 3

LEFT_WRIST

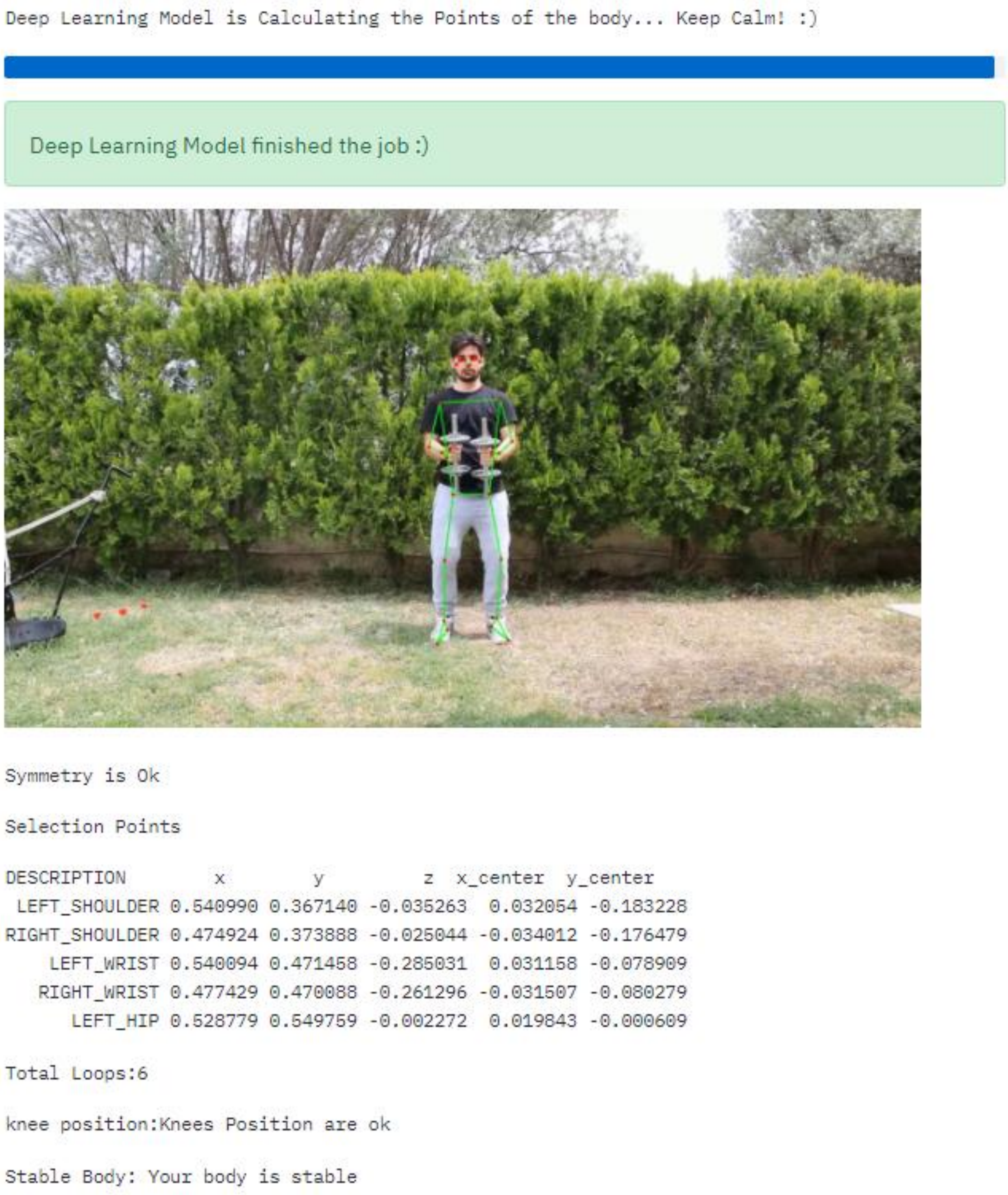
Select Point 4

RIGHT_WRIST

Select Point 5

LEFT_HIP

Figure 28:Hand Stretches Turns with Dumbbells Part 1

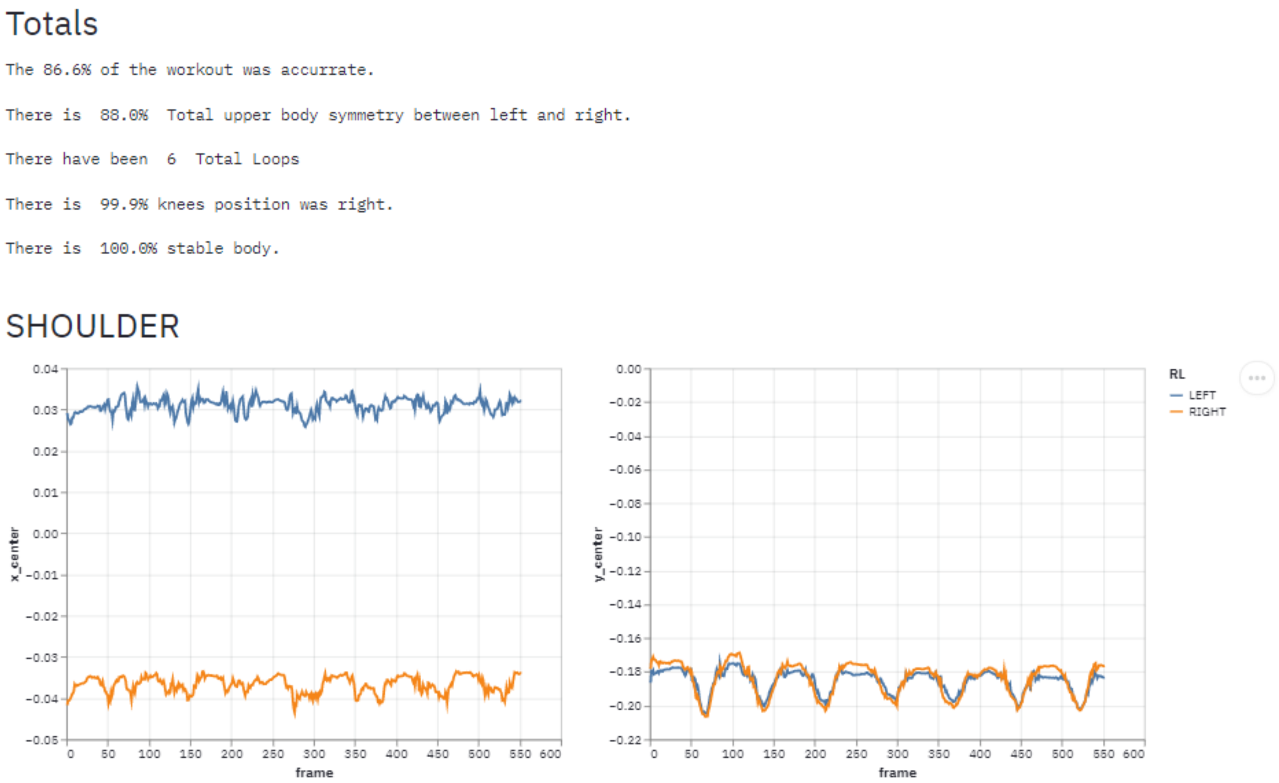


```
Deep Learning Model is Calculating the Points of the body... Keep Calm! :)
```

Deep Learning Model finished the job :)

```
Symmetry is Ok

Selection Points

DESCRIPTION        x         y          z  x_center  y_center
 LEFT_SHOULDER 0.540990 0.367140 -0.035263  0.032054 -0.183228
RIGHT_SHOULDER 0.474924 0.373888 -0.025044 -0.034012 -0.176479
    LEFT_WRIST 0.540094 0.471458 -0.285031  0.031158 -0.078909
   RIGHT_WRIST 0.477429 0.470088 -0.261296 -0.031507 -0.080279
      LEFT_HIP 0.528779 0.549759 -0.002272  0.019843 -0.000609

Total Loops:6

knee position:Knees Position are ok

Stable Body: Your body is stable
```

Figure 29:Hand Stretches Turns with Dumbbells Part 2



Totals

```
The 86.6% of the workout was accurrate.

There is  88.0%  Total upper body symmetry between left and right.

There have been  6  Total Loops

There is  99.9% knees position was right.

There is  100.0% stable body.
```

SHOULDER

Figure 30:Hand Stretches Turns with Dumbbells Part 3

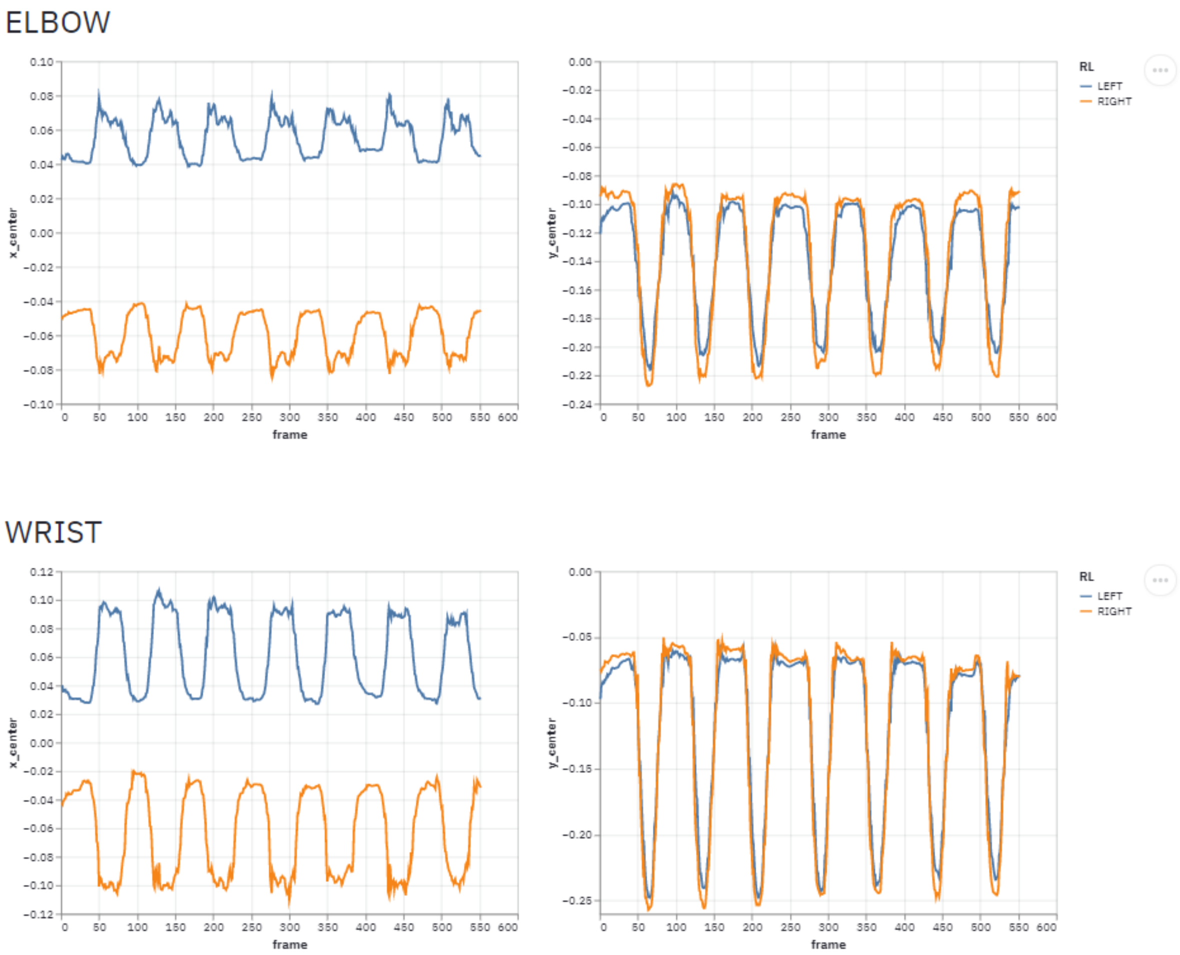


Figure 31:Hand Stretches Turns with Dumbbells Part 4

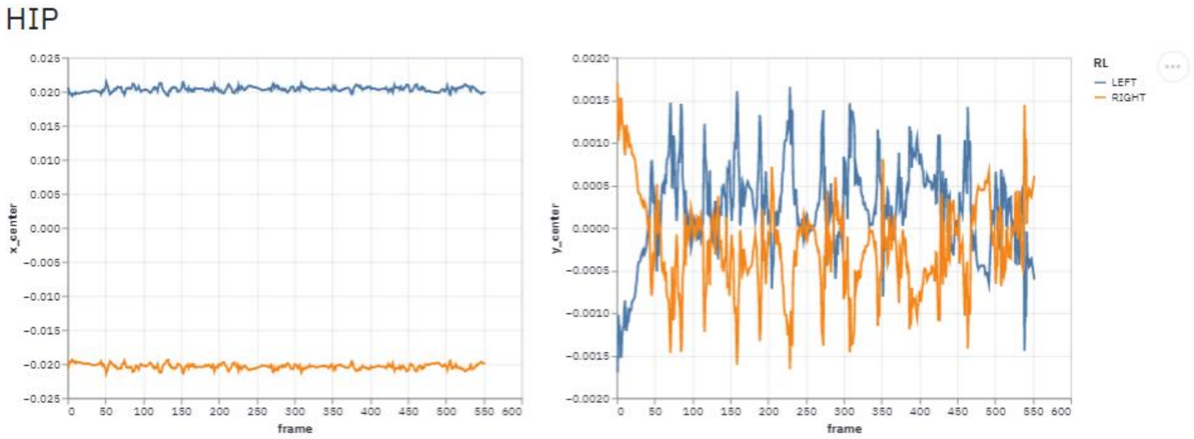


Figure 32:Hand Stretches Turns with Dumbbells Part 5

5.2 Future Work

After the implementation of this app there are many thinks that anyone can do in order create more features in the list below there are some of the features that we would like to see:

1. Live calculations of the checks while performing the exercise.
2. To be able to run algorithms for any position of the camera and not just in the front.
3. To add more workouts.
4. Mobile app

References

[1] "World Health Organization," [Online]. Available: https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/publications-and-technical-guidance/noncommunicable-diseases/stay-physically-active-during-self-quarantine. [Accessed June 2021].

[2] "BBC," [Online]. Available: https://www.bbc.com/worklife/article/20200504-covid-19-update-quarantine-home-workouts-during-coronavirus. [Accessed June 2021].

[3] R. Santos, "NATIONAL INSTITUTE FOR SPACE RESEARCH," [Online]. Available: http://www.lac.inpe.br/~rafael.santos/Docs/CAP394/WholeStory-Iris.html. [Accessed November 2020].

[4] k. P. Murphy, Machine Learning, Massachusetts: Massachusetts Institute of Technology, 2012.

[5] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, o'reilly, 2019.

[6] "IBM," [Online]. Available: https://www.ibm.com/cloud/learn/unsupervised-learning.

[7] "Deepsense," [Online]. Available: https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/. [Accessed May 2021].

[8] J. Patterson and A. Gibson, Deep Learning, o'reilly , 2017.

[9] "TensorFlow," [Online]. Available: https://www.tensorflow.org/resources/models-datasets. [Accessed December 2020].

[10] G. V. Houdt, C. Mosquera and G. Nápoles, "A review on the long short-term memorymodel," *Artificial Intelligence Review volume,* pp. 5929-5955, 2020.

[11] "HumanEva Dataset," [Online]. Available: http://humaneva.is.tue.mpg.de/. [Accessed May 2021].

[12] "Human3.6M," [Online]. Available: http://vision.imar.ro/human3.6m/description.php. [Accessed May 2021].

[13] "MPII Human Pose Dataset," Max planek institut informatik, [Online]. Available: http://human-pose.mpi-inf.mpg.de/. [Accessed May 2021].

[14] "PoseTrack," [Online]. Available: https://posetrack.net/. [Accessed May 2021].

[15] "COCO Keypoint Detection," [Online]. Available: https://cocodataset.org/#home. [Accessed April 2021].

[16] "LSPe - Leeds Sports Pose Extended," [Online]. Available: https://dbcollection.readthedocs.io/en/latest/datasets/leeds_sports_pose_extended.html. [Accessed May 2021].

[17] S. Leonid , A. O. Balan and M. J. Black, "HUMANEVA: Synchronized Video and Motion Capture Dataset," *Springer Science+Business Media, LLC 2009,* pp. 4-27, 2009.

[18] G. W. Taylor, L. Sigal, D. Fleet and G. Hinton, "Dynamical Binary Latent Variable Models for 3D Human Pose Tracking," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2010.

[19] X. Zhou, Q. Huang, X. Sun, X. Xue and Y. Wei, "Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach," *IEEE International Conference on Computer Vision (ICCV), 2017,* pp. 398-407, 2017.

[20] C. Ionescu, D. Papava, V. Olaru and C. Sminchisescu, "Human3.6M: Large Scale Datasets and," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 36, pp. 1325-1339, 2014 .

[21] M. Andriluka, L. Pishchulin, P. Gehler and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 3686-3693, 2014.

[22] K. Sun, B. Xiao, D. Liu and W. Jingdong , "Deep High-Resolution Representation Learning for Human Pose Estimation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 5693-5703, 2019.

[23] V. Belagiannis and A. Zisserman, "Recurrent Human Pose Estimation," *IEEE 12th International Conference on Automatic Face & Gesture Recognition*, pp. 468-473, 2017.

[24] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks," *Proc. ACMM,* 2015.

[25] M. Fieraru, A. Khoreva, L. Pishchulin and B. Schiele, "Learning to Refine Human Pose Estimation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* pp. 205-214, 2018.

[26] F. Zhang, X. Zhu and M. Ye, "Fast Human Pose Estimation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 3517-3526, 2019.

[27] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall and B. Schiele, "PoseTrack: A Benchmark for Human Pose Estimation and Tracking," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 5167-5176, 2018.

[28] I. Akhter and M. Black, "Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1446-1455, 2015.

[29] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition,* 2014.

[30] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 43, no. 1, pp. 172-186, 2021.

[31] "TensorFlow," TensorFlow, [Online]. Available: https://www.tensorflow.org/lite/examples/pose_estimation/overview. [Accessed December 2020].

[32] "3D Human Pose Estimation Python," OpenVINO, [Online]. Available: https://docs.openvinotoolkit.org/latest/omz_demos_human_pose_estimation_3d_demo_python.html. [Accessed April 2021].

[33] "Pose Mediapipe," Mediapipe, [Online]. Available: https://google.github.io/mediapipe/solutions/pose.html. [Accessed February 2021].

[34] V. Bazarevsky, I. Grishchenko, T. Zhu, F. Zhang, M. Grundmann and K. Raveendran, "BlazePose: On-device Real-time Body Pose tracking," *CVPR Workshop on Computer Vision for Augmented and Virtual Reality,* 2020.

[35] "Canon," [Online]. Available: https://www.canon.gr/cameras/eos-2000d/. [Accessed May 2021].

[36] I. Fatouros and I. Chatzinikolaou, Weight Training, Telethrio, 2012.

[37] "Python," [Online]. Available: https://www.python.org/. [Accessed December 2020].

[38] "Pandas," [Online]. Available: https://pandas.pydata.org/. [Accessed December 2020].

[39] "Numpy," [Online]. Available: https://numpy.org/. [Accessed December 2020].

[40] "Streamlit," [Online]. Available: https://streamlit.io/. [Accessed December 20200].