CM2007 - Project

Fitness Activity Tracking

Rebecca Bonato and Benjamin Darçot

November 1, 2023

1 Introduction

This project aims to apply AI techniques on videos of people performing fitness' exercises. In particular, it is possible to split the project in two tasks. First of all, a neural network has been trained to classify videos according to the type of exercise a person is performing. Then, videos concerning the same exercise have been classified according to the 'fitness accuracy': in other words, the purpose is to distinguish when an exercise is performed correctly, partially correctly or incorrectly.

2 Data Analysis

2.1 Dataset description

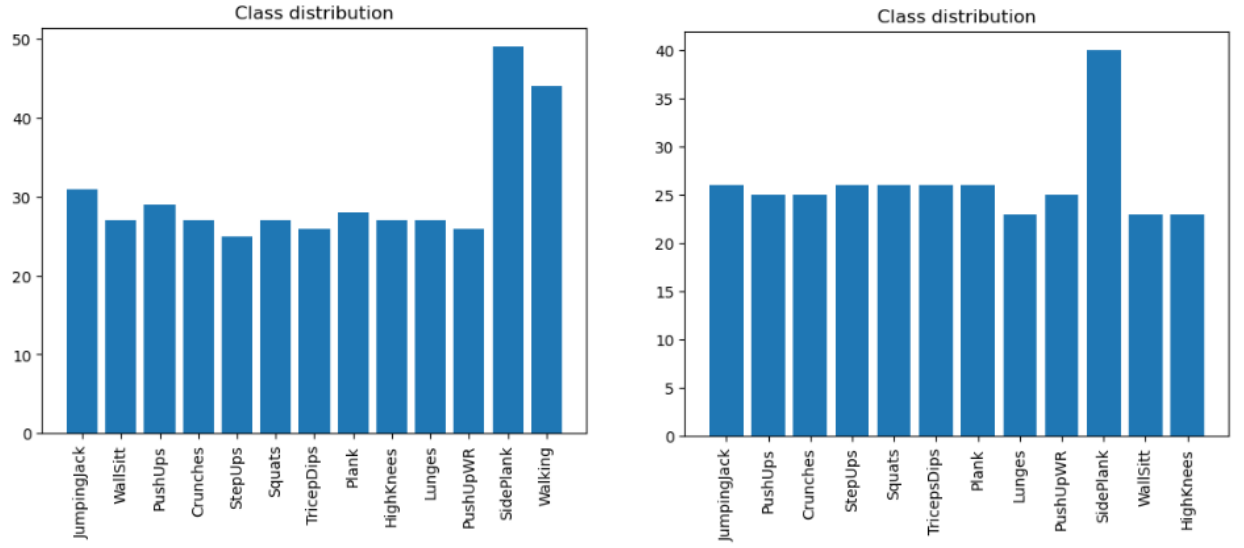
The dataset used is composed of 53 videos in which different people perform a complete training session. It is important to specify that resolution and dimension of videos differ from each other, as well as the accuracy with which exercises are performed. In the file 'Annotations.csv' there are indications about how to split some videos (the first 27 ones) in 'single-exercise videos' and the corresponded labels are provided: these videos will be used for the training and the validation of the model. On the other hand, the remaining 26 videos are used as a testing dataset. Therefore with the best model obtained in the previous step, a classification will be made on this unseen dataset to assess the generalisability of the model. Videos used for training and validation and for the final test classification are shown in Table 1. There are 13 exercises in which the videos are classified: it is a multi-class classification problem.

Training and Validation	**Video to classify (Test)**
1, 3, 4, 5, 6, 7, 8, 10, 11, 12,	31, 32, 33, 34, 35, 36, 37,
13, 14, 15, 16, 17, 18, 19,	38, 39, 41, 42, 43, 44, 45,
20, 21, 22, 23, 24, 25, 26,	46, 47, 48, 49, 51, 52, 53,
27, 28, 29, 30	54, 55, 56, 59, 60

Table 1: Videos used to train the model in the classification task and testing videos classified by the best model obtained.

2.2 Data Cleaning

The first step consists in a process of data cleaning. The 'Annotation.csv' file is characterized by some mistakes (such as incorrect labels association or imprecise or wrong time of beginning/end of the video) that have to be manually corrected. In addition, the time defining each single activity is converted from minutes to seconds in order to facilitate the loading of the dataset, further in the study. Those corrections are made through the code '1_Annotations_to_dataframe.ipynb' and are saved in the 'Annotations_cleaned.csv' file. In addition, the videos not present in the 'Annotations.csv' file are manually labelled and will constitute the testing dataset as mentioned before. These new labels are saved in the same format as the first annotations in the file 'Annota tions_test.csv'. One can also note that the code '1_Annotations_to_dataframe.ipynb' investigates the class distribution in the two datasets (Figure 1).



(a) Class distribution on the training and validation set

(b) Class distribution on the test set: the label 'Walking' is missing

Figure 1: Labels distribution between dataset and test set

Since working with videos is computationally expensive, once videos have been cut in smaller videos (let's call them 'new videos') containing just one type of exercise each, 17 body joint's coordinates are extracted in each frame. This allow us to work with time series saving a lot of memory. This process is done through the 'Create_lighter_dataset.ipynb' file and different steps are enumerated below:

1. Each video has been converted in order to have a frame rate equal to 15 fps which make the different videos more consistent with each other.

2. According with the information contained in 'Annotations_cleaned.csv' and 'Annotations_test. csv' videos have been cut: each of the new video is characterized by only one label that corresponds to the exercise that is performed in it. The problem of inconsistency in video duration was solved by taking into account only the first 150 frames of the long videos and repeating the shorter videos in a loop until 150 frames were reached. In the end, each new video has a duration of 10s (150 frames at 15 fps).

3. For each new video, 17 key points of the body are extracted in each frame, giving a total of 34 coordinates per frame (2 per key point). This extraction is made using MoveNet [2]. The model offers two possible modalities of body joints extraction: lightning and thunder. Both have been investigated and thunder data have finally been used because of its better performance. The extracted coordinates are already normalized with respect to the image size; thus they are in the range [0,1].

4. Data are saved in a data-frame for each new video and the correctness of the coordinates extraction has been checked as shown in Figure 2. Thus, at the end, for each new video, 34 time series of length 150 are obtained by the extraction of the 34 coordinates in the 150 frames.

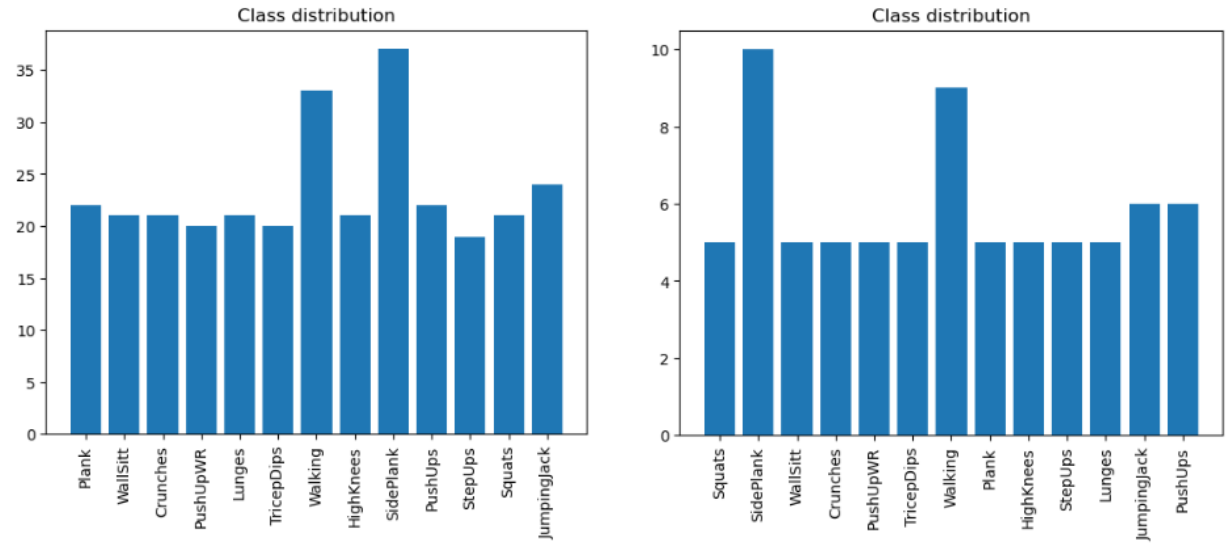
Figure 2: The process used to asses the correctness of the joints extraction. In correspondence of each joints a white square of 20X20 pixels is visible. The frame has been randomly extracted and belongs to video '30 - 7 Minute Workout Full Video.mp4'

At the end of this process we obtain two data sets of time-series: the train and validation set derived by the 'Annotation_cleaned.csv' file and the test set derived from 'Annotation_test.csv' file.

3 Classification

3.1 Data Preparation

The first aim of the project consists in classifying each exercise with respect of the type of fitness activity performed. Thus, the train and validation set was split in such a way that the training set represents 80% of the videos and the validation set 20%, while maintaining a balance in the distribution of the classes (Figure 3).



(a) Class distribution on the training set

(b) Class distribution on the validation set

Figure 3

The number of classes - the number of fitness activities in which a video can be classified - are 13: the task consists in a multi-class classification. Labels have been converted from categorical to binary vectors through Hot-Encoding technique: the choice comes from the fact that there is not a specific order between them.

3.2 Model and parameters

The next step consist in the design of the model. Different combinations between CNN, LSTM and dense layers have been tried in order to find the one more suitable for the task. Models tested were different between each other but they were characterized by some mandatory prerequisites enumerated below.

1. The input layer is characterized by a shape equal to [batch size, number time-step, number features] where the number of time-step is equal to 150 and the number of features is equal to 34 (17 body's joints; each of them is characterized by two time-series: x and y coordinates).

2. The last layer is a Dense layer with 13 neurons for the number of classes. The activation function chosen is 'softmax'. It is useful to normalize the outputs and to convert them into a

probability value for each class that sums to 1. The class with the higher probability is the one predicted.

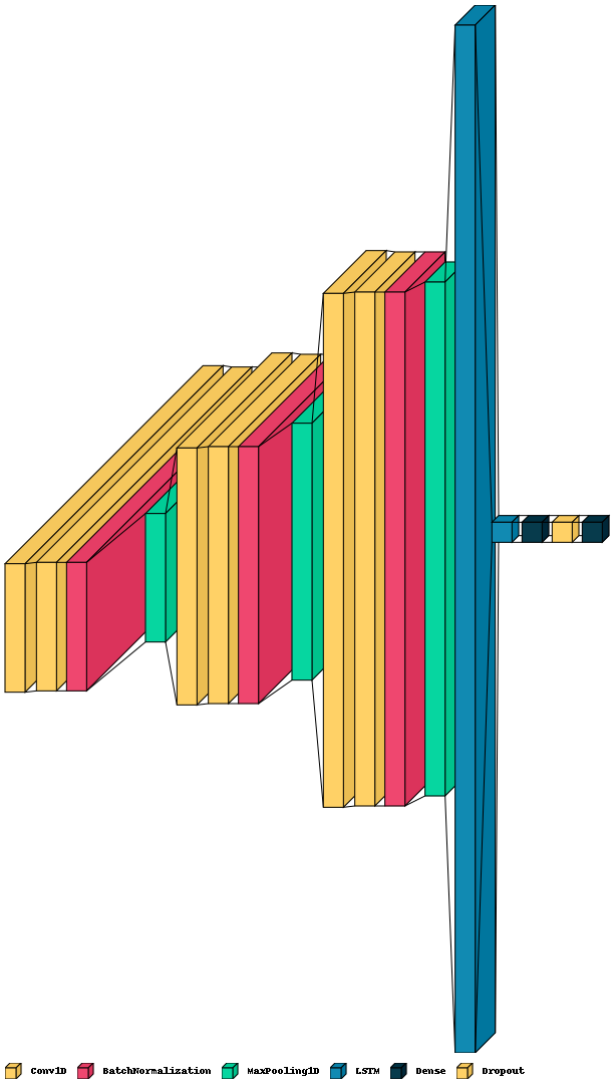


Figure 4: Best network reached after tuning parameters [3]

The model with the best performances has then been selected, after tuning parameters, and saved. Its structure is shown in Figure 4 and described below.

1. Firstly, three convolutional blocks have been inserted. Each of them is characterized by 2 convolutional layers, a batch normalization layer and a max-pooling layer with pool_size equal to 2. In each block the number of neurons of the convolutional layers are respectively 32, 64, 128. This part is largely derived from the encoder of a U-net network.

2. In sequence, two LSTM layers have been inserted (256 and 128 units respectively). Recurrent dropout is added to avoid overfitting (0.2 and 0.1 respectively).

3. A dense layer with 128 neurons is then added before the final layer with a dropout rate equal to 30%. The last layer is a dense layer as well, as explained above.

As loss function, 'categorical crossentropy' has been used during the training process while the optimizer chosen is Adam with a learning rate equal to $5e^{-5}$. The batch size selected is equal to 8 and the metric with which the results are evaluated is the accuracy score. These parameters are summarized in Table 2.

Table 2: Chosen parameters

Training Parameters	
Parameter	Value
Optimizer	*Adam*
Learning rate	$5e-5$
batch size	8
loss function	*CategoricalCrossentropy*

3.3 Results and Discussion

The process of training and validating the neural network has been tracked and it is shown in Figure 5. From these trends, it is possible to claim that, even if the performance on the validation set are much more variable with compared to the one on the training set (i.e. oscillating), the training process is working properly and there is not real overfitting. Once convergence is reached, the accuracy on the validation set is in average around 93% with a peak of 97%. The model with the best accuracy has been saved to classify data belonging to the testing set.

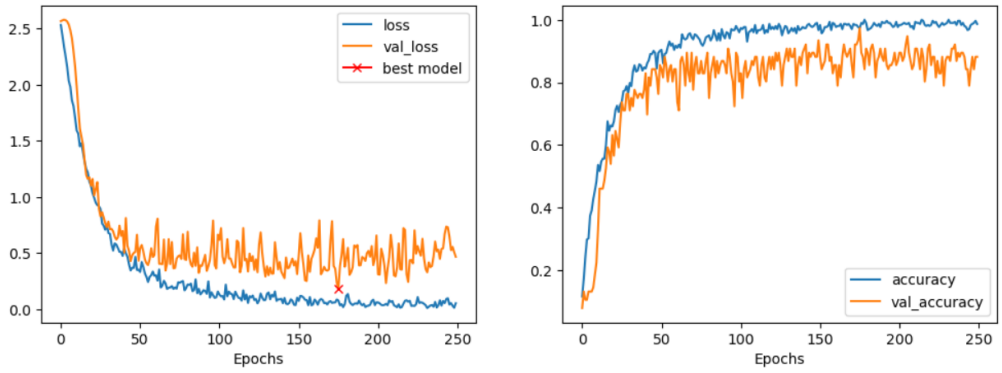


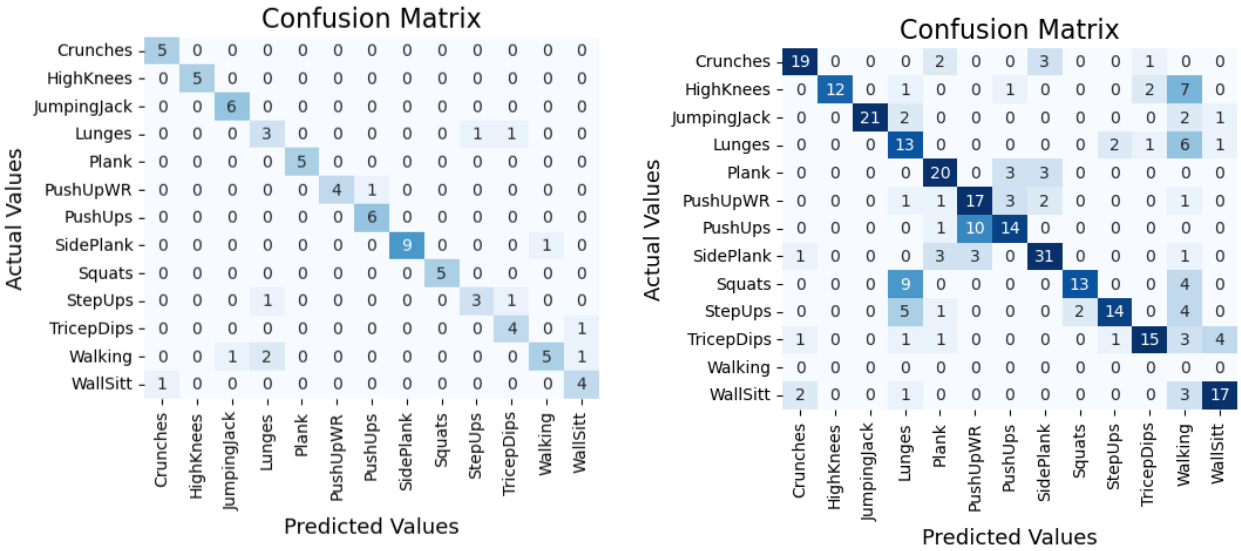
Figure 5: In the figure is shown the training process of the model in terms of loss and accuracy

Due to the high number of classes, the accuracy score is not enough to assess the quality of the classification and to enhance drawbacks. Confusion matrices, shown in Figure 6, help to improve the understanding of the performances. By looking at the one from the validation set 6a, it's possible to see that the diagonal is clearly enhanced, which means that the classification is well performed, and that the higher error rate is associated with the label 'walking'. This is reasonable: the quality of the data labelled as 'walking' is much worse since people were not performing this as an exercise but it was just an activity to move from an exercise to another.

On the testing set's confusion matrix 6b the diagonal is clearly visible as well even if the number of misclassifications is much higher than before. The first explanation is that these data

are new to the model and thus a decreasing in the performance was expected. Furthermore, there are several raw videos with a very low quality in the test set. Indeed, some issues were found: the presence of more than one person in the video leading consequently to an incorrect output from the Movenet, low quality of exercise accuracy - that could create misclassifications between 'HighKnee' and 'Walking'- and low quality of the video shooting itself.

Despite these drawbacks which highlight the dependence on the quality of input data, the model is clearly able to perform the task for which it has been trained and to provide consistent results.



(a) Confusion matrix on Validation set

(b) Confusion matrix on Test set

Figure 6

4 Fitness Accuracy Evaluation

The second purpose of this project is to evaluate the fitness accuracy of the different activities performed in the videos, i.e. to define how well the exercises are performed. For this purpose, as no annotations are available, the task has been treated as an unsupervised machine learning problem. Thus, the idea here is to build three different clusters each representing a certain level of accuracy such as: good fitness accuracy, medium fitness accuracy and poor fitness accuracy.

4.1 Method

In a first step, a single activity is selected; in this study it is the squat. From there, a K mean algorithm for time series [1] was chosen to build the 3 clusters composed of the different squat videos. The K-means algorithm is well known and widely used to perform similarity clusters based on a distance between the data (often Euclidean). However, as the data for this task are made up of 34 time series (for each coordinates) of length 150 (for the number of frames), the distance between the data is not obvious to define, especially as it is important that the algorithm takes into account both the time series of each video relative to each other as well as the temporal dimension of each series. Indeed, for an exercise to be well performed, it is necessary that at each time the position is correct, i.e. that the joints are all well placed in relation to each other, but also that

from a dynamic point of view the positions are correct, i.e. that the same joint does not undergo any abnormal change in time. It is therefore to take into account these two effects that the K mean for time series was used with the 'Dynamic Time Warping Distance'.

4.2 Results and Discussion

The 52 squat videos were indeed clustered into 3 different clusters 0, 1 and 2 of size 8, 27 and 17 respectively. The details of these clusters are performed and displayed in the code '4_Clustering.ipynb'. The fact that no cluster is empty shows that the algorithm has managed to find similarities and differences between videos of the same activity, however these differences are not necessarily based on fitness accuracy as it would have been intended.

Indeed, after verification it seems that the main factors that change from one cluster to another are the camera behaviour and the positioning of the subject in the frame. Cluster 0 seems to group all squat videos where the camera is fixed. Cluster 1 seems to group the videos where the camera turns around the subject and where the subject goes out of the frame at some point. And finally, cluster 2 seems to group videos where the camera is moving and where there are cuts in the editing of the video.

Therefore, the result is not convincing and does not seem to be the expected one, even if it is important to note that not being fitness specialists it is difficult for us to evaluate the fitness accuracies in the different videos. These failing results seem to be due to a too great diversity in the videos and maybe also to a lack of relevant features for clustering. Indeed, it seems that with standardised videos in all datasets the differences observed in the clusters could not be present anymore. Furthermore, a better selection of features may be necessary; although the position of the joints seems to be very relevant for this task, it may not be sufficient and other features could be extracted such as joint angles for example.

5 Conclusion

Finally, the classification of sports activities using a neural network is reasonably accurate and easy, even with relatively low quality data. However, the clustering technique for fitness accuracy is very weak and does not seem to be suitable for this task or at least for this kind of data set. Furthermore, this project shows the importance of data processing through the crucial steps of data cleaning and feature extraction from the videos. The cleaning allows a better quality of data while the feature extraction allows a drastic reduction of the memory needed for the realization of the model afterwards and is essential when studying such a heavy database.

AI and especially machine learning are techniques that can be applied in almost all fields nowadays, even in sports technology. This project is part of this approach and demonstrates the importance of these new technologies.

References

[1] Alexandra Amidon. *How to apply K-means clustering to time series data*. Aug. 2021. URL: https://towardsdatascience.com/how-to-apply-k-means-clustering-to-time-series-data-28d04a8f7da3.

[2] *MoveNet: Ultra fast and accurate pose detection model*. https://www.tensorflow.org/hub/tutorials/movenet?hl=it.

[3] *visualkeras for Keras/TensorFlow*. https://github.com/paulgavrikov/visualkeras.