Physical Exercise Form Correction Using Neural Networks

Cristian Militaru
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
cristian.militaru@student.utcluj.ro

Maria-Denisa Militaru
Babes-Bolyai University
Cluj-Napoca, Romania
maria_denisa1999@icloud.com

Kuderna-Iulian Benţa
Babes-Bolyai University
Cluj-Napoca, Romania
benta@cs.ubbcluj.ro

ABSTRACT

Monitoring and correcting the posture during physical exercises can be a challenging task, especially for beginners that do not have a personal trainer. Recently, successful mobile applications in this domain were launched on the market, but we are unable to find prior studies that are general-purpose and able to run on commodity hardware (smartphones). Our work focuses on static exercises (e.g. Plank and Holding Squat). We create a dataset of 2400 images. The main technical challenge is achieving high accuracy for as many circumstances as possible. We propose a solution that relies on Convolutional Neural Networks to classify images into: correct, hips too low or hips too high. The Neural Network is used in a mobile application that provides live feedback for posture correction. We discuss limitations of the solution and ways to overcome them.

ACM Reference Format:
Cristian Militaru, Maria-Denisa Militaru, and Kuderna-Iulian Benţa. 2020. Physical Exercise Form Correction Using Neural Networks. In *Companion Publication of the 2020 International Conference on Multimodal Interaction (ICMI '20 Companion), October 25–29, 2020, Virtual event, Netherlands.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3395035.3425302

1 INTRODUCTION

Doing exercises the right way is not an easy task. If the user wants serious results he/she has to learn the correct way of training. The wrong posture or technique is not just ineffective, it can also lead to serious injuries. Having a personal trainer reduces these risks, but it is expensive and gyms can become overcrowded.

An Artificial Intelligence mobile solution would allow users to exercise from any place. There is a growing interest in the commercial sectors for an artificial fitness trainer on mobile devices, proven by 3 successful (i.e. with positive user reviews) apps [12][1] [3] [5]. Also, for basketball, there are 2 artificial coach apps [11] [9]. The application developers do not disclose their proprietary algorithms. Despite the market interest, other studies do not focus on mobile devices, therefore this work can be considered as a small step towards shrinking the gap between industry and research.

[1]The Onyx iOS application has 4.9 out of 5 stars, with more than 3000 ratings at the moment of writing

ICMI '20 Companion, October 25–29, 2020, Virtual event, Netherlands

ACM ISBN 978-1-4503-8002-7/20/10…$15.00
https://doi.org/10.1145/3395035.3425302

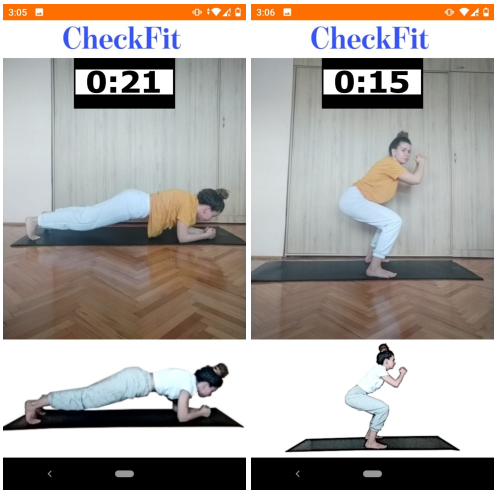

Figure 1: Screenshots of the mobile application: left - the Plank exercise, right - Holding Squat. The top part displays a timer and live camera images, while the bottom side provides a picture with the correct posture. The user receives live audio feedback on how to correct the hips position.

The main contribution of this work is a labeled dataset with 2400 images with Plank and Holding Squat (balanced across classes)[2]. These images are collected indoors, by the authors, and labeled by the authors with the help of a fitness trainer. The labels are categories: hips too low, correct[3], or hips too high. Furthermore, it presents a solution using Convolutional Neural Networks and a benchmark for evaluating the robustness to different scenarios (backgrounds, outfits etc.). Also, a mobile application is developed that gives real-time feedback on how to improve posture. The tested solution is based on open-source libraries (TensorFlow and Android). The results are discussed and directions for improvement are provided.

2 RELATED WORK

To the best of our knowledge, there is no work in the literature for general-purpose physical exercise correction using commodity hardware (smartphones). Other studies require sensor devices such as Kinect 3D and a PC to detect whether the posture is correct or not. They address a small group of users who own or are willing to purchase such devices. By requiring only a smartphone, a larger

[2]The dataset and source code are available at https://github.com/cristipiticul/check-fit.
[3]Exercise correctness is a complex concept. In this work, we only consider one common mistake: hips too low or too high. The system can be extended for multiple posture deviations (e.g. hunching the back in Holding Squat), but some mistakes cannot be caught from camera images (e.g. breathing too fast).

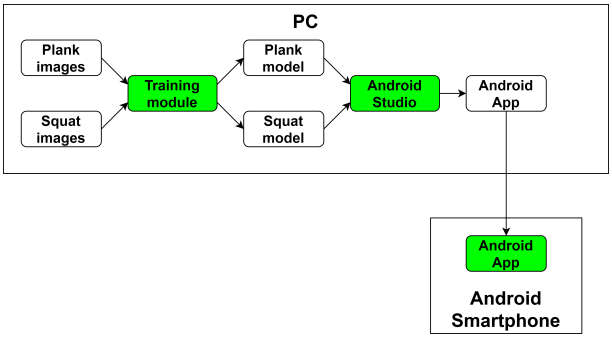

Figure 2: Application development diagram: starting with image datasets for each exercise, train Neural Network models and embed them in the Android application.

population is considered. Moreover, it does not require setting up a designated space for exercising, a mobile device can be moved and used anywhere, including outdoors.

The work in [13] presents a method to detect wrong posture of the Plank exercise in an image, using a two-step procedure: first identify the key body joints (shoulders, hips, feet etc.) using an undisclosed algorithm, then feed the key body joints to a small neural network. The system is compared to a rule-based algorithm, which checks whether the angle of the back is in a fixed interval, and the neural network improves accuracy by 6%. The detection is run on a desktop machine. Although the neural network is small enough to be run on a mobile device, the joint detection algorithm is unspecified and might not be applicable to this use-case.

Other works require additional hardware. [8] focuses on rehabilitation exercises and allows patients to continue their recovery from home. The technique requires a Nintendo Wii Remote© (motion sensor that communicates via Bluetooth) to be held in the hand, or attached to the moving limb. Some works use a Kinect 3D sensor to acquire the joint positions (shoulders, hips etc.), then classify the posture by applying either a rule-based algorithm [16] or Neural Networks [6].

Additional studies use smartphone devices' movement sensors to assess the quality of the performed exercises[10] [7]. The mobile device is fixed on fitness equipment. Therefore, they assume the user has fitness equipment in his/her home, or he/she is at the gym. Instead, our task is more general and challenging because it relies on camera images. This allows evaluating a wider range of exercises, without requiring any fitness equipment.

3 APPLICATION DETAILS

The application uses images from the frontal camera. At every 8 seconds, it takes a picture and classify it using a Neural Network in one of three classes: hips too low, correct[3], or hips too high. Then the application provides feedback: verbal indications on how to improve the hips position (e.g. "Your hips are too low, please lift them up").

Fig. 2 shows the application development process. First, we acquire a dataset of 2400 labeled images. The images are organized into folders by exercise (Plank, Holding Squat), by label (hips too low, correct[3], hips too high) and by tested scenario (background, outfit etc. - see next section). Then the images are passed to the training algorithm to produce the models. The training can be from scratch (models initially have random weights), or from a model trained for another task, technique called transfer learning. The main advantage of transfer learning is that it requires less training data and time. The layers closer to the input are kept as they are, because they are extracting basic features from the image, such as round shapes or combinations of straight lines. The layers closer to the output are trained from scratch, those being the ones responsible for doing the classification, knowing the basic features. Finally, the resulting models (one for Plank and one for Holding Squat) are placed in the "assets" directory of the Android application. The app uses these files to classify live camera pictures using TensorFlow Lite [1].

Table 1: Image attributes variation results on validation data

Attribute	*Accuracy*	*Loss*	*AUC*	*Observations*
Background	0.33	1.28	0.65	Underfitting
Lighting conditions	0.61	1.22	0.73	Overfitting
Outfit	0.73	0.49	0.84	Overfitting
Distance	0.80	0.40	0.93	
Exercise variation	0.81	0.60	0.84	Overfitting
Mirroring	0.85	0.30	0.91	
Different person	0.90	0.25	0.91	
Angle	0.95	0.20	0.97	

4 EVALUATION

The application is evaluated in two ways for different purposes:

- Assess the robustness, by varying image attributes (background, lighting conditions, outfit etc.) and measure the classification performance;
- Determine the training parameters (number of epochs, Neural Network architecture etc.) that achieve the best results.

In both tests, for evaluating the classification performance the following metrics are used: accuracy, loss and area under ROC curve (AUC).

4.1 Analyzing Image Attributes

Table 1 presents the results for each image attribute variation. Only Plank images are used in these tests. For each attribute, 3-4 variations are selected. One is held-out for validation, while the others are used in training. For example: *background* - 3 backgrounds are chosen (2 for training and 1 for validation). For each background, we take a fixed number of pictures, distributed evenly in each category (e.g. 40 correct, 40 too high, 40 too low).

When evaluating an attribute, all the other attributes are not altered: for the *background* variable, the outfit is the same in all pictures. An exception is the angle, which is varied by up to ±15° from the right angle in order to ensure some variance in training data.

A *MobileNetV2*[14] neural network is trained for 10 epochs using transfer learning (pre-trained model on ImageNet dataset). The architecture is using specialized layers that require less computation, parameters and memory. The authors use 19 bottleneck blocks, each of which has two 1x1 2D convolutional layers to combine

Figure 3: Backgrounds used for evaluations: a CNN is trained using pictures with the 1st and 2nd backgrounds, and validated using images with the 3rd background

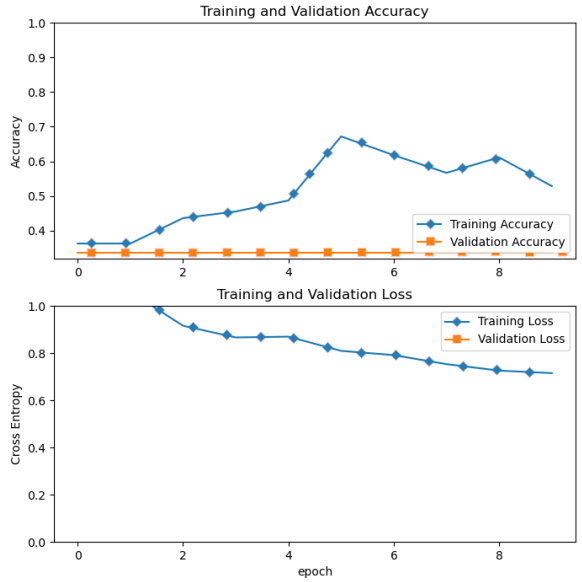

Figure 4: Background variation: results of training on 2 backgrounds (312 images), and testing on a 3rd background (114 images). Plots from top to bottom: accuracy and loss during training. Plots show signs of underfitting.

channel information, and between them a Depthwise Separable Convolution (1 filter per channel).

The number of epochs was determined using trial and error, where more epochs would not increase the performance, and fewer epochs would produce inaccurate models.

The results can be grouped into 3 categories: underfitting, overfitting and successful learning.

Fig. 4 displays the performance of the model when varying the **background** (see also Fig. 3). Plots reveal the **underfitting** problem, since the accuracy is smaller than 0.55 for both training and validation (also $AUC = 0.65$). To overcome this issue, a possible solution is to add more images to the dataset with more backgrounds. Another improvement could be to also allow the first layers of the network to be re-trained, giving the network more capacity for accomplishing the task.

Fig. 5 displays a gap of 0.23 in accuracy and 0.5 in loss between training and validation performance (also $AUC = 0.84$). Therefore,

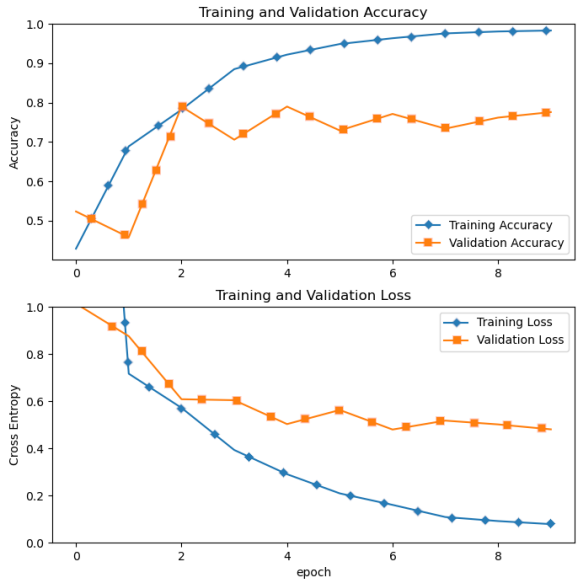

Figure 5: Outfit variation: results of training on 2 outfits (409 images), and validating on a 3rd outfit (213 images). Plots from top to bottom: accuracy and loss during training. Plots show signs of overfitting.

in the case of **outfits**, the model is **overfitted**. Other attributes with similar results are: **lighting conditions** - training: 187 images with natural daylight and medium light, validation: 102 images with artificial light; **exercise variation** - training: 120 images with regular Plank, validation: 42 images with one leg lifted. For these attributes, adding more variation and images in training set could improve the results.

The training process is **successful** in Fig. 6 (also $AUC = 0.97$). Similar to **angles**, there are no overfitting or underfitting signs when varying: **distance** - training: 186 images taken from a distance of 2m, validation: 60 images at 1m. The latter images cannot contain the whole body, so we leave out the non-essential parts for classification (i.e. feet and head for Plank exercise); **mirroring** - training: 186 images of Plank on one side, validation: 67 images after the person rotated 180°; **different persons** - training: 210 images of one person, validation: 55 images with another person. Even though the network is trained successfully, the dataset contains only small variations of the parameters and no corner-case situations. For example, in the case of different persons, both of them had a similar aspect (body shape) and similar outfits. In the case of angles, only ±15° variations of the right angle are tested. Therefore it is worth varying these parameters more for a higher robustness.

4.2 Training Parameters

Table 2 displays different Neural Networks performance, trained on subsets of the Plank exercise dataset (1257 images), for different number of epochs. The dataset is divided in 5 parts, out of which one is used for validation. The table shows that performance increases

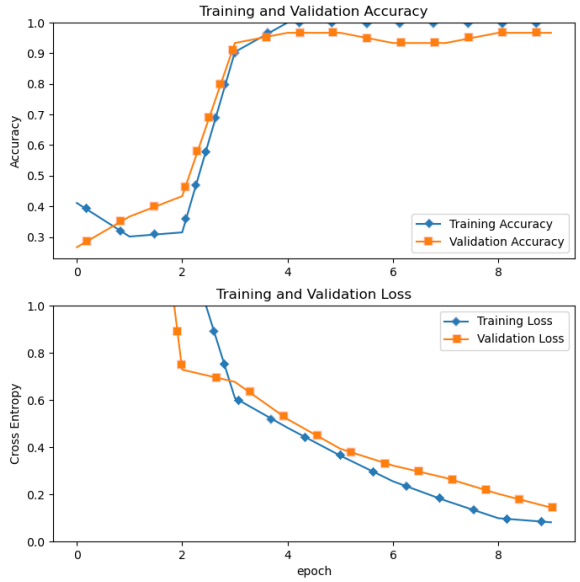

Figure 6: Angle variation: results of training on 3 angles (255 images), and validating on a 3rd angle (74 images). Plots from top to bottom: accuracy and loss during training. Plots describe a successful learning.

Table 2: Validation results depending on no. images and no. epochs

	10 epochs		*20 epochs*		*30 epochs*	
	Acc.	Loss	Acc.	Loss	Acc.	Loss
251 images (1/5 split)	0.62	0.61	0.85	0.40	0.65	0.44
503 images (2/5 split)	0.83	0.13	0.93	0.21	0.62	0.81
754 images (3/5 split)	0.65	0.80	0.98	0.05	0.97	0.06
1006 images (4/5 split)	0.96	0.37	0.95	0.20	0.99	0.04

as the number of images and number of epochs increase, with a few exceptions. This probably happens because the training algorithm is non-deterministic: weights are initialized with random values, and images are shuffled before each epoch.

Total training time depends on dataset size and number of epochs. From our measurements[4], 1 epoch requires 36s for the largest dataset (4/5 split).

Another important factor is the newtork architecture. All the above tests have been using the *MobileNetV2* architecture. We compare it to two other architectures designed for mobile/embedded devices: *MobileNet*[4] and *Mobile NASNet*[15]. All models are trained for 30 epochs on the 3/5 split dataset from previous test. The achieved accuracy is greater than 0.98 for all architectures, but there is a difference in training times:

- *MobileNetV2* - 28s per epoch (14min total);
- *MobileNet* - 42s per epoch (21min total);
- *Mobile NASNet* - 59s per epoch (30min total).

[4]Timings are recorded for training using an Intel Core i3-3220 (no GPU)

5 DISCUSSION

Although the evaluation section show promising results, our proposed method has significant limitations.

Firstly, it cannot handle dynamic (moving) exercises. To solve the problem for dynamic exercises, further work could either keep the current architecture and add logic for handling detection results on multiple frames, or use time-aware neural networks, for example Long-Short Term Memory architectures.

Secondly, the robustness to some environment attributes (background, lighting conditions, outfit) is low. This is probably due to insufficient training data. Expanding the dataset should improve the robustness. Another direction would be to use an existing pose estimation model [2] to identify the positions of key body joints (shoulders, hips, knees etc.) in the image. Then feed positions to the classifier.

In comparison to prior work, we expect the accuracy of our system to be lower because of not using specialized hardware sensors for determining the pose. But similar accuracy should be achievable by employing a larger dataset and a better processing algorithm. The main advantages of our solution are: being able to provide real-time feedback to end-users and offering flexibility inherent to mobile applications.

6 CONCLUSION

This paper presents a labeled dataset for training Artificial Intelligence models to help correct posture in static fitness exercises. It proposes a solution based on Convolutional Neural Networks and evaluates the performance under different environmental conditions (background, camera angle, distance etc.). The trained network is embedded in a mobile application. Our work reveals some challenges and solutions to overcome them.

Apart from the improvements suggested by the previous section, further work could enhance the flexibility of the training process in order to make the solution more customizable by end-users and trainers. Instead of embedding the models inside the application, one can use a database and create a web service for users to upload their own images and train new models. Another challenge comes when considering rehabilitation patients that cannot perform an exercise correctly. Then the system would need to adapt through the recovery phases, at first being loose, and becoming more demanding in time, similar to how a human personal assistant would behave.

ACKNOWLEDGMENTS

We thank Fitness Trainer Bogdan Bora for correcting our postures and for the help provided in labeling images.

REFERENCES

[1] [n.d.]. *TensorFlow Lite.* https://www.tensorflow.org/lite/ (visited: 2020/07/02).
[2] [n.d.]. *TensorFlow Lite Pose Estimation.* https://www.tensorflow.org/lite/models/pose_estimation/overview (visited: 2020/07/02).
[3] Exer Labs Inc. [n.d.]. *Perfect Plank by Exer.* https://www.exer.ai/ (visited 2020/07/02).
[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *Computing Research Repository (CoRR)* abs/1704.04861 (2017).
[5] Kaia Health Software Inc. [n.d.]. *Motion Coach by Kaia™.* https://www.kaiahealth.com/ai-platforms/motion-coach/ (visited 2020/07/02).

[6] Y. Liao, A. Vakanski, and M. Xian. 2020. A Deep Learning Framework for Assessing Physical Rehabilitation Exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28, 2 (2020), 468–477.

[7] P. S. Madanayake, W. A. D. K. Wickramasinghe, H. P. Liyanarachchi, H. M. D. M. Herath, A. Karunasena, and T. D. Perera. 2016. Fitness Mate: Intelligent workout assistant using motion detection. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*.

[8] Oscar Marin-Alonso, Daniel Ruiz-Fernández, Antonio Soriano, and Joaquin D. Garcia-Perez. 2013. Use of Multilayer Perceptron vs. Distance Measurement Methods for Classification of Exercises in Telerehabilitation. In *Natural and Artificial Models in Computation and Biology*. Springer Berlin Heidelberg, Berlin, Heidelberg, 171–180.

[9] MaxOne. [n.d.]. *M1 SmartCoach.* https://maxone.ai/smartcoach/ (visited 2020/09/26).

[10] A. Möller, L. Roalter, S. Diewald, J. Scherr, M. Kranz, N. Hammerla, P. Olivier, and T. Plötz. 2012. GymSkill: A personal trainer for physical exercises. In *2012 IEEE International Conference on Pervasive Computing and Communications*. 213–220.

[11] NEX Team Inc. [n.d.]. *HomeCourt.* https://www.homecourt.ai/ (visited 2020/09/26).

[12] Onyx Inc. [n.d.]. *Onyx: Home Workout.* https://www.onyx.fit/ (visited 2020/07/02).

[13] Alexander Pfyffer. [n.d.]. *Detecting Wrong Posture of the Plank Fitness Exercise using an Artificial Neural Network.* https://github.com/Vollkorn01/Deep-Learning-Fitness-Exercise-Correction-Keras (visited 2020/07/02).

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4510–4520.

[15] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. 2019. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[16] Alexander Vybornyi, Vladimir Rozaliev, and Yulia Orlova. 2017/12. Controlling the correctness of physical exercises performance. In *Proceedings of the IV International research conference "Information technologies in Science, Management, Social sphere and Medicine" (ITSMSSM 2017)*. Atlantis Press, 233–237.