https://doi.org/10.22364/htqe.2022.80

DEVELOPMENT OF THE APPLICATION TO PROMOTE DAILY PHYSICAL ACTIVITIES IN ADOLESCENTS

Edgars Oļehnovičs[1], Artūrs Laizāns[1], Aija Kļaviņa[2], Juris Porozovs[1], Viktors Veliks[1], Kārlis Freivalds[1]

[1] University of Latvia, Latvia

[2] Latvian Academy of Sport Education, Latvia

ABSTRACT

Many people, especially adolescents, use mobile applications to control their physical activities and reduce sedentary behaviors. While a large variety of sports applications are available, detection of the activity type is a new research area. There are also specific methods how to measure and collect data or control the type of sports activities. Literature research shows that analysis of the current sports application market is evolving in a fast speed. Some authors show that neuron networks can help detect and analyze specific data of sports activities using sensors or mobile phone.

The aim of this research was three-folded: 1) to develop sports application for a simple exercise to promote physical activity in adolescents, 2) to study the abilities of artificial intelligence in a sports application context, and 3) to create an application for detecting and counting everyday physical activity.

Methodology: two applications were created, (1) for data collection, and (2) for everyday usage. Many deep learning models and their parameters were tested.

Results: application prototype to detect one simple physical activity (squats) using neuron networks was created, one neuron network model to detect and count activity squats was made. Neural network models for walking and jumping rope were created.

Conclusions: it is possible to use sports application for adolescents to promote their physical activity. The neuron network model can be used in mobile phone, while the use of this model is dependent on various factors.

Keywords: *mobile application, adolescents, physical activity, deep learning, neuron networks*

Introduction

The aim of this research is to create an app to motivate adolescents to become physically active, our tasks are to research artificial intelligence possibilities in the sports application context and create an application to detect activities and count them.

United Nations admits insufficient physical activities are in the 4th place in the risk of death. In the time of the pandemics the risk rises even higher (Meyer et al., 2021). Physical activities are important in promoting the development of adolescents, they reduce the risk of heart diseases, cancer etc (Hallal et al., 2006; Barkleya et al., 2020). There are many methods for reducing the risk of being physically inactive: making games, using special personnel as well as creating mobile sports applications (Barkleya et al., 2020; Luo, He, 2021). There is ongoing research specifically in the neuron networks field to create sports applications, that can detect physical activities.

There are many examples of sports applications, for example, BunnyBolt, Runtastic. The effect of application usage on adolescents to promote physical activity in general is positive. Specific tasks and games have been developed to promote physical activity for adolescents. It is found that applications can create positive effects like improving motivation to do physical activities and making fun. Competition can occur between adolescents, but it is not the main factor (Barkleya et al., 2020, Keung et al., 2013; Klenk et al., 2017).

Artificial intelligence is an actively researched field when activity needs to be counted and/or detected. Birmingham university researchers have tested many artificial intelligence methods. Good results were received by methods that are not deep neuron network: k-nearest neighbor, support vector machine, shallow neuron network. The best results were achieved with the deep learning method. Researchers used mobile phones to collect data, and if it was needed, participants held phone in hand to analyze and collect data. Best accuracy was 0.974 which they got from deep neural network (DNN) (Huang et al., 2022).

Another example is from the University of Antioquia Columbia. Researchers created an application to detect 19 different everyday activities and 15 special sports falling activities and used deep neural network to create the application. Activities included for example running, sitting, jumping, get into and off the car. The used model use RCNN-LSTM (recurrent convolutional – long short term memory) neural network (Sucerquia et al., 2016).

In our project we made an approach to count and quantify physical activities using YOLO (you only look once) model. There are many ways how to calculate results from obtained data. Various optimization methods could be used. It is possible to create logical schemes to get results. There are models which allow playing games. There are developed models for calculating many different moves. It is possible to plan when and what could be done, for different obtained data. The hardest part is to make the model learn from given data and results. Learning model better reacts to known data, but for new data new model needs to be trained, then good output results could be achieved (Russell & Norvig, 2021).

Deep learning

A theoretical analysis of deep neural networks is performed in this subsection. Deep learning could be defined as when there are many layers, they are connected with each other, and they influence each other. The core element of the neural network is a neuron or base function also called perceptron. A base function is a mathematical function. Examples of base functions are sigmoid, ReLU, tanh etc. Neuron networks are connected similarly to brain, of course, connections can be in different shapes and layers and their number depends on each model' s needs. There are different types of deep neural networks (Russel & Norvig, 2021; Krohn, 2020). Convolution neural network (CNN) is used to minimize data processing. Convolution result is defined as data and filter multiplication. In the convolution network, there are given input data, and after the convolution procedure, there is less information at the output (Burkov, 2019; Russel & Norvig, 2021).

Using neural networks multiplying and filtering of data are used at the beginning. Then we sum it up, and additionally, we can add an extra sum (bias). Filters can be different on different data. The filter is also called the kernel. For data borders, there can be padding as well (extra data zeroes or similar). Kernel size usually is 3x3 and it can be 2×2 or other. In our model 3x3 filter size is used. And there is also parameter speed (also called strides). It means how fast the filter is moving around data. If strides are 2 then data on the next layer is double less, if 3 then triple less. In our model 1 or 2 strides are used. Using parameter stride we make a layer smaller. Flatten layer is used to remove complexity and we get only result in data after flattening layers. There can be more than one flattened layer (Burkov, 2019; Webster, 2022; Foster, 2019).

A recurrent neuron network can remember previous data. Information about previous steps is used to generate the next steps (Coşkun et al., 2017). Long short-term memory (LSTM) is usually used in recurrent networks to remember previous data. It is also needed to avoid losing the gradient (Coşkun et al., 2017; Balouji et al., 2018).

YOLO (You only look once) model is used to create a new deep – learning network to detect and count physical activity. Our YOLO model consists of 8 convolution layers and the filter size is 3×3 and the strides value is 2, as well batch normalization and flatten layer, and dropout coefficient of 0.2 (Laizāns, 2021). The YOLO model is a deep convolution neuron network. YOLO deep learning model is usually used to detect objects in images fast. YOLO network consists of 63 convolutional layers and uses 3×3 and 1×1 filter size (Redmon & Farhadi, 2018). In our research we are using a deep convolution neuron network that is based on the YOLO model. After training model was able to detect and count

physical activities. Three different models based on the YOLO approach were created for squats walking and jumping rope.

Model is trained when positive values are given to train it. It is being done many times (epochs) so model becomes trained. We use deep neuron network to train model. Model's training result is checked, when test is made on validation data. When model is successful it is tested in life.

To make an application that can detect activities deep learning method is the most appropriate method. The deep learning model is made based on YOLO because it can detect gyroscope and accelerometer data which is changing over time. To train the model data is given to the model then the model is tested. After algorithm detect activity successfully (approximately 90%) testing result model is put on the mobile phone within the application. The application then can be used to detect and count activity and give adolescents feedback about how much activity is done. If observed results didn't meet selected parameters, algorithm training cycle performed again.

During physical activity intervence with adolescent's research team needs instruments that meet the following requirements, first allows to control over how exercises are done by participants, second team members have possibility to change exercises during the intervences time, third access recorded data from their own data storage that is independent of any commercial vendors. The main difference from other projects is that our model can count activity and show result on how much of this activity is being done. Therefor more relevant feedback could be given when specific activities are being done.

Methodology

Development of the application

To develop an application for promoting the physical activity of adolescents first we need to collect data, then data need to be used to create a deep learning model and then create an application using a trained model. There are two parts to application creation. One application is needed to collect data, the second application is necessary to use the trained model in practice. HARCollect application is used to collect data. HARCollect application is used to get data and label them. It shows arrows and timeline, where user can label data on video recording. When labeling is done, data is saved on Firebase server. Two mobile phones are needed to collect and then label collected data. Data collection with one phone is not possible. 2 phones simultaneously are needed. HARCollect (see Figure 1) application is used in the following steps.

1. Installing of application.
2. Registration of users.

3. Connecting 2 phones.
4. Collecting of data.
5. Labeling of data.
6. Uploading of data.
7. Training of data in Colab with Tensorflow etc.

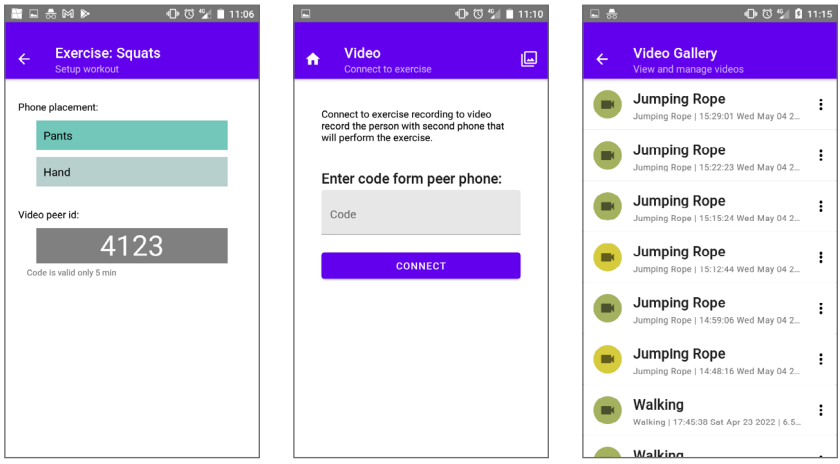

Figure 1. HARCollect application in use, left exercise and phone placement, middle code synchronization dialog and right video list screen

The second application is Time to move (Laiks kustēties). It was used to test squat detection in a controlled environment. The application showed a count of squats activity e. g. 1, 2, 3. for squat activity. If further developed It could be used to give users feedback about specific activity for a user. Application is made to detect and count activity. It is integrated with Google fitness and Firebase server. The application in the time of writing is in debugging prototype stage.

Google Colab and Firebase usage

Google Firebase is used to get data from users, register users as well as to host application. Accelerometer and gyroscope data are collected in Firebase as well as labels when squats are started and ended. Users can send data remotely to server. They can choose what data are sent. Google Colab is needed to create csv data files from accelerometer and gyroscope data for model training. Then deep learning model is created in Colab. Colab can read data from Firebase.

Process steps:

1. Data collection in the application HARCollect, user registration.
2. Data uploading in Firebase.

3. Data labeling in Colab.
4. Data csv file creation in Colab.
5. Csv files reading in Colab.
6. Model creation.
7. Model parameter testing.
8. Model export.
9. Model usage in life by application Time to move.

Data quality needs to be checked before data are used in model creation.

Used data and data collection

Data for application training was collected from 8 adolescents 10–14 years old and 12 adults 23–50 years old. The application's main purpose is to use it with adolescents during physical intervention. Therefor used model and algorithms must be adapted for daily use by adolescents and during application testing they provide feedback about usability of the developed application. Following physical exercises were used for application squats, jumping rope and walking. For algorithm training at least half an hour of squats data was created and 15 minutes of jumping rope and walking data were used.

Data collection takes place when 2 phones are connected, then data need to be labeled and send to the Firebase server. The application is paired with Firebase server and automatically gets data that is sent. Every user is registered, so it is possible to omit data which are not usable. For data collection Android phones are used. During data labeling both phones are synchronized, otherwise, data labeling cannot happen.

Data labeling is happening by manually using time frames with positive data e. g. when there is squat action it is positive value and is labeled. When person stands or doing something else it is negative value and is not labeled. Then labeled data is used to train model.

Created models

Different models were created but did not give results good enough. Convolution networks are the most popular method for a given problem and were used to create models. The technologies used to create neuron networks are Google Tensorflow, Keras, and Colab.

The first created model was a neuron network with 3 convolution layers. Data were collected at the speed of 20 points per second in 6 axis gyroscope and accelerometer. 2.5 seconds long windows were used for data analysis. The input data layer size was 50x6. The model consisted of 3 convolution layers and the max pooling parameter between convolution layers was used. At the end of the layers were ReLU activation, softmax activation, and 2 fully dense layers were used. 75% of the data was used

for training and 25% data for validation when the model was trained. When the model was trained after 30 epochs it could not learn more. This model was not able to count activities. To detect and count activities simultaneously different network was needed.

Convolution layers were used to try to detect and count activities simultaneously. Idea was that different length windows could be used. The time for every activity (squats) was detected to be 4 seconds long, but working with different data sizes was not possible. This kind of network was not created at all, because of different data and TensorFlow abilities.

The next model was repetition counting the 3K network. All data was fixed in 4 second long fragments. If needed padding of zeroes is used. The network contained 2 convolution layers, a dropout layer with a coefficient of 0.2. In application this kind of network was with very low precision and was not used. After these tests with negative results, the YOLO approach was used to create the next models as shown in sections trained models and results.

For walking and jumping rope activities different models were created before good results were got. The main problems were data fragment length and data labeling. When labeling was not precise enough, models showed low accuracy.

Trained models

Our deep learning model is created using the YOLO model, then it is adjusted to create a new model for physical activity detection and counting. It is important that this adjusted model can count and detect activities simultaneously (see Figure 2).

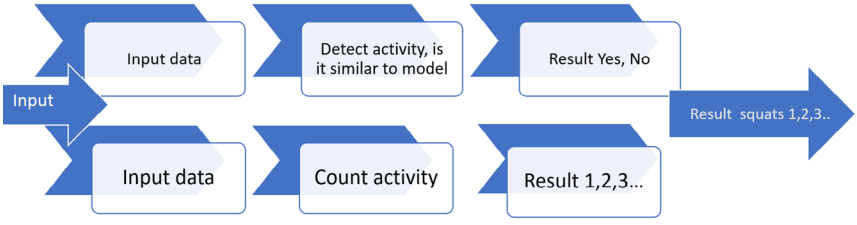

Figure 2. Adjusted model's data flow

A new model is created based on squat data for squat detection. Participants make labels when the squat is started and ended. Input data is gathered from the HARCollect application (see Figure 1). When the model is trained the model can give results to the Time to move application. It is very important to do this labeling right. The best results will be when the activity is supervised because labeling is not an easy task. Data is labeled

and then sent to train the model. To get the model maximum precision, model parameters are changed with Adam optimizer for deep learning model optimization. Adam is a built – in optimizer in Keras for easier use for developers. In Figure 3 you can see a squat data from the gyroscope and accelerometer. The arrow shows the squat middle point. That is where squat is happening in 4 seconds long windows. Also, 8 second long window was tested, but it gave a less accurate result. Computational model optimal results were observed in 4 seconds long data window for squat and walking and 1 second for jumping rope.

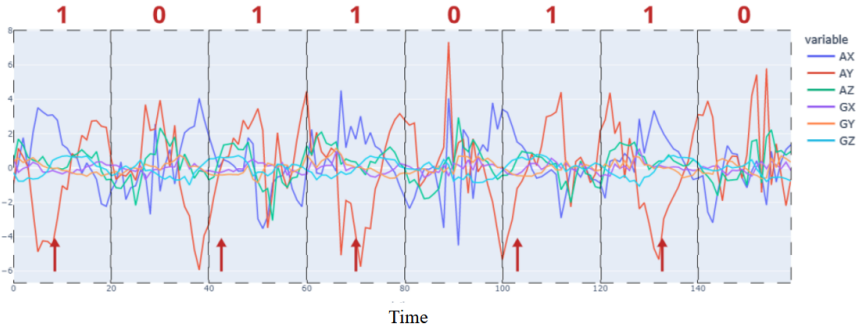

Figure 3. Labeled mobile phone accelerometers in 3-d coordinate (AX, AY, AZ) and gyroscope (GX, GY, GZ) sensors data

Data analysis is started with input data. Data is divided into 4 second fragments. Before that, there are labels on squats. After that 4 second fragments are labeled with 1 or 0. 1 means there is a squat middle point, zero means there is no squat middle point. To adjust the model there is needed hyperparameter optimization. For squat detection 4 seconds best fit for squat activity because it is the approximate length of squat activity as well when the model is learned it shows the best result.

Table 1. Different tested parameters when model is evaluated

Parameters	**Tested values and parameters**	**Value and parameter information**
Convolution filter 1	16, 32, 48, 64, 72	Convolution blocks are in pairs
Convolution filter 2	16, 32, 48, 64, 72	Convolution blocks are in pairs
Convolution pairs	1, 2, 3, 4, 5, 6	How many pairs are optimal
Fully connected layer	No, 10, 25, 50, 100	Tested if fully connected layer more than two is needed
Dropout layer	0.2, 0.3, 0.4	Tested optimal dropout percentage
ReLU	0, 0.05, 0.1, 0.15	Leaky ReLu coefficient. 0 means no Leaky.

To do that TensorFlow plugin HParams is used. Optimization is made by Adam optimizer. In Table 1 there are different parameters, which were tested in different combinations.

To do parameter checking faster some parameters were taken as good as possible and then tested another parameter. After checking extra dense layer is not needed. If one parameter showed the best results, then it was used to check the next parameters, to find the optimal combination at the end, see Table 1. Figure 5 shows parameters that were tested additionally. Figure 4 shows parameters from table 1.

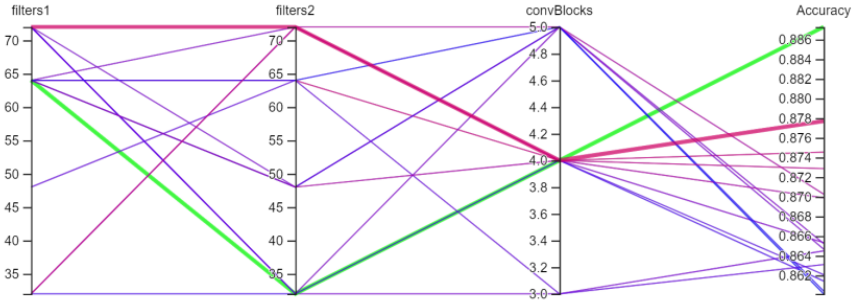

Figure 4. Different convolutions blocks and filters parameters changes and observed accuracy when tested

The best loss function was a mean square error. Convolution layer window size is fixed – 3. Training data versus trained data is 75/25%. If the parameter was found as good enough, then it was used in the next training as optimal. To train the network 60 epochs are enough for model training. The network changes every time when it is tested, even if data don't change. There is also no necessity for an extra dense flattened layer. When tested 64 filter1 counts gave the best results, 72 filters2 gave the best results, 5 convolution blocks gave the best result and the resulting accuracy was 90%. When a model is trained by different parameters it creates result weights, which are needed to detect and count activity (see Figure 2, Table 1, Figure 4).

Results

For the walking activity, best created model got 92% precision when trained 100 epochs.

The deep learning model consists of 2 convolution blocks divided into 1 pair. The first convolution block has 64 filters and strides size 1, and the second block has 72 filters and strides size 2. All convolution filter (kernel) size is 3. In every pair second layer adds extra zero padding. After

convolution, there is a batch normalization layer. In activation layer is used Leaky ReLU function and the dropout layer with a coefficient of 0.2. In Figure 5 you can see how the model is trained.

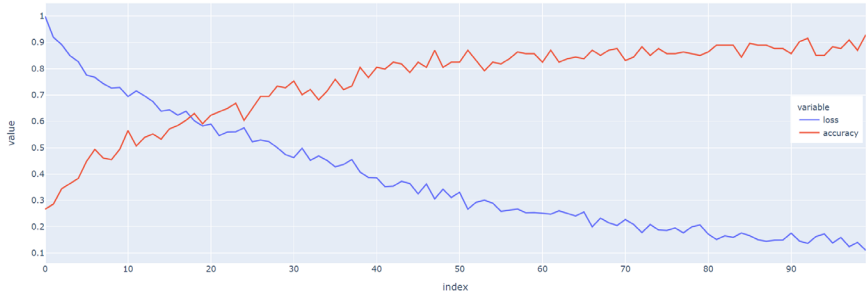

Figure 5. Deep learning model training for walking activity with function loss and observed accuracy

For the jumping rope activity, best created model got 93% precision when trained 100 epochs. The deep learning model consists of 10 convolution blocks divided into 5 pairs. The first convolution block has 64 filters and stride size 1, the second block has 72 filters and stride size 2. All convolution filter (kernel) size is 3. Padding for both pairs is the same. After convolution there is a batch normalization layer. In activation layer is used Leaky ReLU function and the dropout layer with a coefficient 0.2. In Figure 6 you can see how the model is trained.

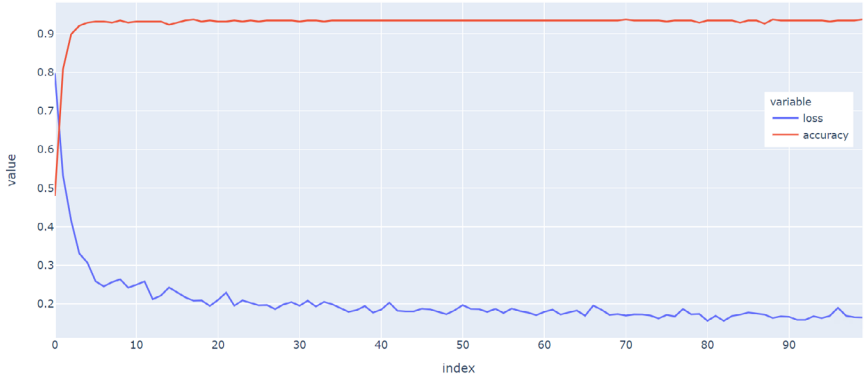

Figure 6. Deep learning model training for jumping rope with function loss and observed accuracy

The deep learning model for squats detection consists of 8 convolution blocks divided into 4 pairs. The first convolution block has 64 filters and stride size 1, the second block has 32 filters and stride size 2. All convolution

filter (kernel) size is 3. In every pair second layer adds extra zero padding. After convolution there is a batch normalization layer. In activation layer is used Leaky ReLU function and dropout layer with a coefficient 0.2. In Figure 7 you can see the model structure.

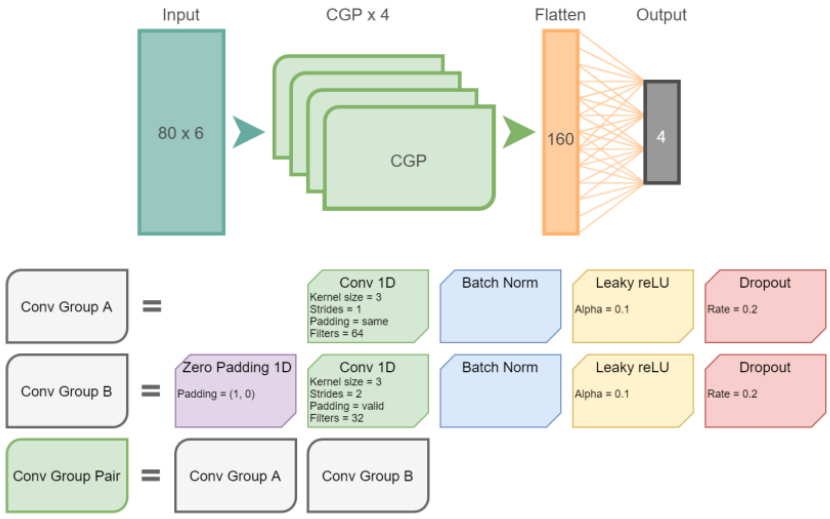

Figure 7. Created model structure for squats detection with used parameters

Trained deep learning model accuracy for squats detection counting is 90%.

The following results were got using squats model: true positive: 98 (76.6%), true negative: 21 (7.4%), false negative: 30 (23.4%), true negative 263 (92.6%)

Trained deep learning model accuracy for squats detection is 94%.

Application when tested in life could count 6 of 10 squats precisely. Two persons were tested. Every person made 10 squat series 10 times. An error was not larger than two in 10 squats.

Created applications when upgraded can be used for adolescents as well as for other interested groups.

The TensorFlow lite model was tested on Qualcomm Snapdragon 845 processor (2017 model) Android 10 operating system. The, model is 213 KB large and 6.8 ms fast.

The model is very dependent on the user and any other factors e. g. right or left pocket.

HAR Collect application is created and the Laiks kustēties (Time to move) application has been tested and is in the development stage.

During creating applications there are the following problems, which must be solved:

1. Long model data training period.
2. Many dependencies when creating an application.
3. Long data collection period.
4. Developers need to know different programming languages and tools: Tensorflow, Keras, Python, Javascript etc.
5. Developers need to know how to use deep learning methods to create a new model.

Discussion

There were a couple of models tested before we got the best results. The main problem with why models did not work was data quality and data labeling quality. At the start of the model creation, there was a challenging phase when there was no working model. After the first model is created later next model creation is easier.

We created 3 models: squat activity – 90%, jumping rope 93%, and walking activity with 92% accuracy.

Charissa Ann Ronao used a convolution network to count activities from mobile phone data. Activities were running, walking up and down the stairs, sitting and sleeping. They got 95% performance accuracy (Ronao & Cho, 2016). Our result was little less than their.

Francisco Javier Ordóñez used Deep convolutional LSTM neuron networks, but the sensors used were through the body. With Deep convolutional LSTM they got 91%, but with baseline CNN 88% (Ordóñez & Roggen, 2016). Our deep convolutional models showed same accuracy rate than their network, and seems this method is better than their baseline model. We got similar results with different convolution layer parameters.

Andrea Soro used one neuron network to detect and different network to count activities. Smart watches on the hand and knee were used. Detection precision was 99% and counting precision was 91% (Soro et al., 2019). Better result of their study could be achieved because of hand watch usage. In our case our results is from phone in a pocket. Sensors movement amplitude is higher in case of hand watch and then better results could be got. Both methods can precisely detect given activity.

Ghanashyama Prabhu used data from the bracelet. In this approach, 10 different activities were tested. Classification precision was 94%, and counting precision was 90%. Two convolution neuron networks were used to gain results (Prabhu et al., 2021). Their result was gained with advanced set of sensors. They used specific local muscular endurance exercises for rehabilitation purpose. They used supervised machine learning models

and deep neuron networks. They found that deep neuron network showed better result than supervised machine learning models. Our result was similar to them, but they used specific devices with extended accelerometer, magnetometer and gyroscope sensor set.

All models compared above and our models have different algorithmical and technical setup but results are more or less similar and got at least 90% precision.

Conclusions

Different methods for physical activity detection are actively worked out in both commercial for using with different trackers applications and academical for new device and applications prototypes. Very promising is physical activity detection and detection of activity peculiarities by deep learning methods. These methods allow to detect small differences between physical exercises for example squats amplitude and count correctly performed activities via deep learning model.

Created application for adolescents could help promote physical activity. Application prototype is used to detect squats, but it needs further development on different activities and testing of adolescents. Our results could be used to further develop application that could give feedback on specific activities to people who is using it in that way promoting activity.

Author Note

This work was supported by the Latvian Council of Science under Fundamental and Applied Research grant Nr. lzp-2019/1-0152.

References

Balouji, E., Gu, I. Y. H., Bollen, M. H. J., Bagheri, A., Nazari M. (2018). A LSTM-based deep learning method with application to voltage dip classification. *18th International Conference on Harmonics and Quality of Power (ICHQP)*, 1–5. https://doi.org/10.1109/ICHQP.2018.8378893

Barkleya, J. E., Leppa, A., Santob, A., Glickmana, E., Dowdella, B. (2020). The relationship between fitness app use and physical activity behavior is mediated by exercise identity. *Computers in Human Behavior*, 108. https://doi.org/10.1016/j.chb.2020.106313

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Amazon

Coşkun, M., Yildrim, Ö., Uçar, A., Demir, Y. (2017). An overview of popular deep learning methods. *European Journal of Technic EJT, 7*(2). https://dergipark.org.tr/en/download/article-file/437659

Foster D. *Generative Deep Learning* (2019). O'Reiley Media, Inc.

Hallal, P. C., Victora, C. G., Azevedo, M. R., Wells, J. C. K. (2006). Adolescent Physical Activity and Health. *Sports Med 36,* 1019–1030. https://doi.org/10.2165/00007256-200636120-00003

Huang, J., Kaewunruen, S., Ning, J. (2022). AI-Based Quantification of Fitness Activities Using Smartphones. *Sustainability*, *14(2),* 690. https://doi.org/10.3390/su14020690

Keung, C., Lee, A., Lu, S., O'Keefe, M. (2013). BunnyBolt: a mobile fitness app for youth *IDC '13: Proceedings of the 12th International Conference on Interaction Design and Children* https://doi.org/10.1145/2485760.2485871

Klenk, S., Reifegerste, D., Renatus, R. (2017). Gender differences in gratifications from fitness app use and implications for health interventions. *Mobile Media & Communication*, *5(2),* 178–193. https://doi.org/10.1177/2050157917691557

Krohn, J. (2020). *Deep Learning Illustrated A Visual Interactive Guide to Artificial Intelligence*. Pearson Education, Inc.

Laizāns, A. (2021). Master thesis. YOLO approach to human activity classification. *University of Latvia*

Luo, W., He, Y. (2021). Influence of sports applications on college students' exercise behaviors and habits: A thematic analysis. *Alexandria Engineering Journal*, *60(6)*. https://doi.org/10.1016/j.aej.2021.03.059

Meyer, S. M., Landry, M. J., Gustat J., Lemon S. C., Webster C. A. Physical distancing ≠ physical inactivity. (2021). *Translational Behavioral Medicine, 11*(4), 941–944. https://doi.org/10.1093/tbm/ibaa134

Ordóñez, F. J., Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors, 16*(1), 115. https://doi.org/10.3390/s16010115

Prabhu, G., O'Connor, N. E., Moran, K. (2020). Recognition and Repetition Counting for Local Muscular Endurance Exercises in Exercise-Based Rehabilitation: A Comparative Study Using Artificial Intelligence Models. *Sensors, 20*(17), 4791. https://doi.org/10.3390/s20174791

Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *Computer Vision and Pattern Recognition.* https://doi.org/10.48550/arXiv.1804.02767

Ronao, C. A., Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications 59*, 235—244. https://doi.org/10.1016/j.eswa.2016.04.032

Russell, S., Norvig, P. (2021) *Artificial Intelligence A Modern Approach Fourth Edition* Pearson Education

Soro, A., Brunner, G., Tanner, S., Wattenhofer R. (2019). Recognition and Repetition Counting for Complex Physical Exercises with Deep Learning. *Sensors. 19(3),* 714. https://doi.org/10.3390/s19030714

Sucerquia, A., López, J. D., Vargas-Bonilla, J. F. (2016) SISTEMIC, Faculty of Engineering, Universidad de Antioquia UDEA. https://github.com/Zeo-shark/Human-Activity-Recognition-using-Mobile-Sensor-Data; http://sistemic.udea.edu.co/en/investigacion/proyectos/english-falls/

Webster, K. (2022) *Getting started with TensorFlow 2*. https://www.coursera.org/learn/getting-started-with-tensor-flow2