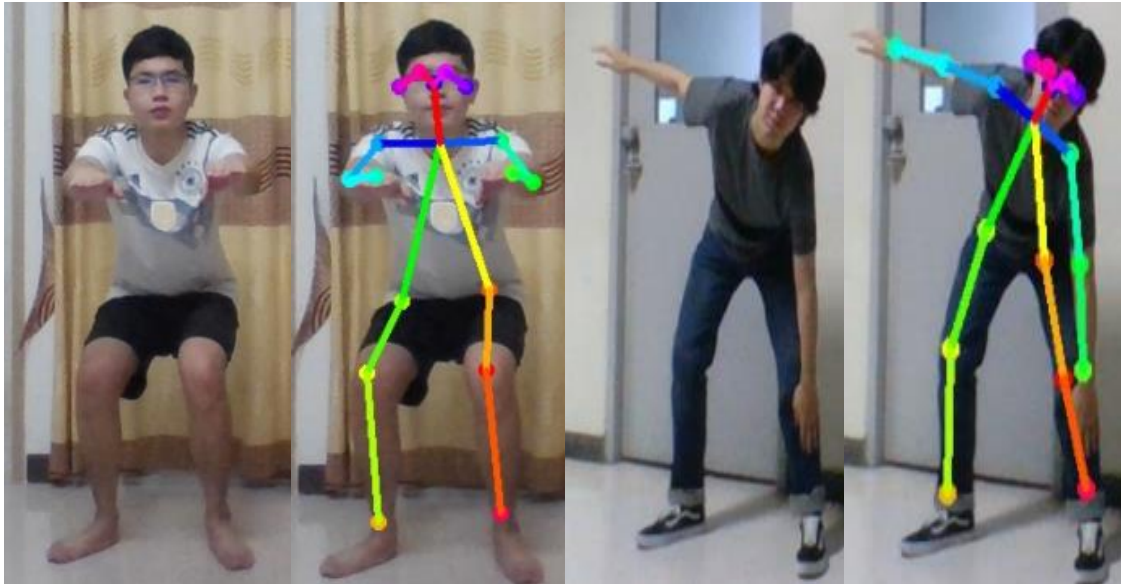


Exercise Movement Counting Program using pose estimation



Group C4

3404 Ms. Jiajia Bai

3410 Ms. Sonia Gautam

3419 Mr. Tunwa Satianrapapong

3422 Mr. Napas Vinitnantharat

3426 Mr. Panithi Suwanno

Teaching Assistant

Saran Khotsathian

Date

14 DEC 2020

Abstract

The objective of this work is to count the human exercise gesture with high accuracy by human pose estimation. It has been in focus of significant augmented reality technology that is widely used at the present and plays a crucial part in many fields. This work concerns on human action recognition focused on tracking and classifying articulated body motion. Such method requires accurate localization of body part, which is a difficult task, particularly under realistic imaging conditions. In this paper, we compare CMU and MobileNet model features for action recognition in each scenario. The application of exercise pattern recognition approach is used for classifying the input images by Multilayer Perceptron method. Training set feature is generated by exercise gesture from trainer who provided advance performance images for further training into TF pose estimation which consists of set, knee touch and squat down. Simulation result shows that high accuracy classifiers with less fault dismissal rate are realizable with classifier algorithm. This project involves less time to identify each image and making it suitable for real time exercise evaluation.

Table of content

Abstract	i
Introduction	1
Background and related work	1
Human pose perception	1
TF-pose-estimation	2
Multi-layer perception	2
cuDNN	3
Tool and Methodology	4
Data set collection	5
Model for pose estimation	6
Classifying Gesture Algorithm	7
Result and Discussion	8
Conclusion	10
Appendix A: Tools specification	11
Appendix B: Original and augmented data comparison	12
References	13

Acknowledgement

We would like to express special thanks of gratitude to our teacher assistant “Sarun Khotsathian” for his able guidance and support in completing our project.

We would like to extend our gratitude to the lecturer “Asst. Prof. Jumpol Polvichai, Ph.D.” for providing us with all facility and truth of invaluable knowledge that was required.

Group C4

Introduction

While exercising and working with different type of training such as squat or knee touch etc. for long length of time, it requires the essential tools to deal with this in order to get rid of counting mistakes during exercise movement. This project makes exercising much easier and encourage users to access more concentration and efficiency while training [1].

Over last two decades, machine learning has transformed the field of computer vision. Deep convolutional networks were successfully applied to learn different vision tasks such as image classification, image segmentation, object detection and many more [2]. Moreover, TF Pose Estimation is one of the most popular bottom-up approaches for human pose recognition, partly because of their well implementation. In activity perception by tracking the variation in the pose over the period of time can also be used in fitness field [3].

Therefore, the combination of machine learning algorithmic model and high prototype exercise gesture information that has been produced will solve the exercise problem without the requirement of superfluous things. For instance, trainer or fitness machine. All of which will change the exercise form and perform in the constructive way in the future.

Background and Related work

The background and related works in action recognition focused on tracking body parts and classification the joint movements. The pose-based approaches, while straight-forward, requires accurate tracking of body parts only, which is an inspiring task to apply this technology more productively and more tangibly [4].

Human pose estimation

A Human Pose Skeleton represents the orientation of a person in a graphical format. Essentially, it is a set of coordinates that can be connected to describe the pose of the person. Each co-ordinate in the skeleton is known as a part (or a joint, or a key point). A valid connection between two parts is known as a pair (or a limb). Note that, not all part combinations give rise to valid pairs. A sample human pose skeleton is shown figure 1.



Figure 1. Left: COCO key point format for human skeletons. Right: Rendered human pose skeletons

TF-Pose-Estimation

TF-pose-estimation is open-source human pose estimation algorithm that have been implemented using Tensorflow [5]. It also provides several variants that have some changes to the network structure of ‘Openpose’ [6] for real-time processing on the CPU or low-power embedded devices which has been developed from real time multi-person 2D pose in key joint. Enabling machine to have an understand people images to detect the 2D pose of multiple people in variety of image. The proposed method uses a nonparametric representation, which is refer to as Part Affinity Fields to learn associate body part individually in the image. This bottom-up system achieves high accuracy and real time performance regardless to the number of varieties in the image. Combined detector not only reduces the inference time compared to running them sequentially, but also maintains the accuracy of each component individually. This work has culminated in the release of OpenPose, the first open-source real time system for multi-person 2D pose detection [7], including body, foot, hand, and facial key points.

Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN) [8]. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training [9]. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable [10]. The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptrons that are organized into layers. An alternative is "multilayer perceptron network".

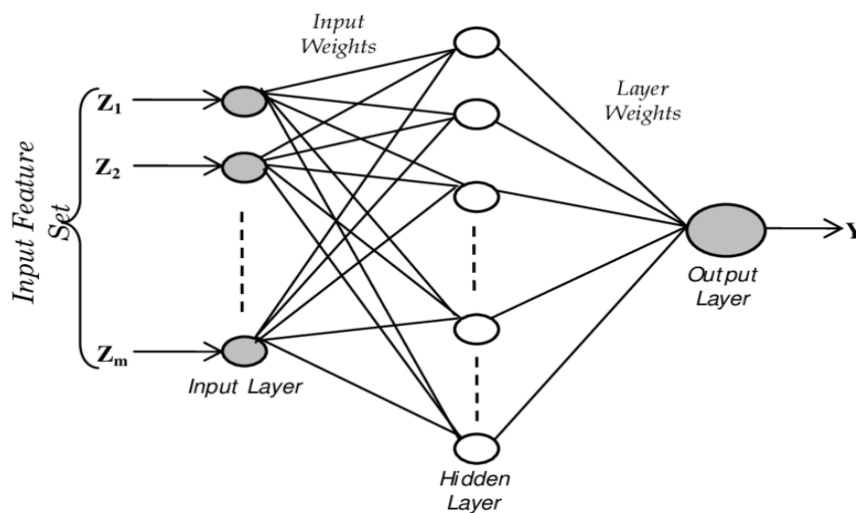


Figure 2. Multi-layer Perceptron architecture with 1 hidden layer

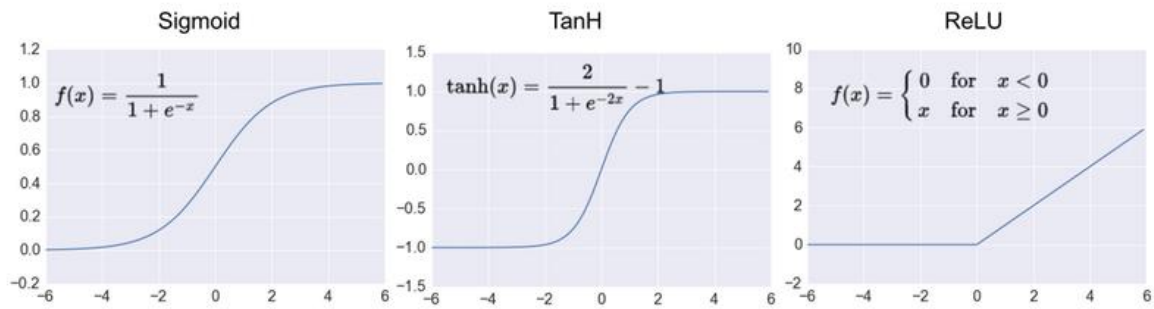


Figure 7. common activation function sigmoid (left), tanh (middle), ReLU (right)

In This project, MLP Architecture that have been used for classifying the exercise gesture which has 2 hidden layers, in each layer have 64 neurons and all neurons have an ReLU as activation function. The Adam optimization have been used for the MLP classifier to adjust the hyper parameter of the model.

cuDNN

The nvidia cuda deep neural network library (cuDNN) is a GPU- accelerated library which provides highly tuned implementations for convolutional graphic. cuDNN is optimized higher performance for AI and computer vision which is used to enhance the ability to render images in every frame better. This is because visual processing unit is used for processing rather than central processing unit only.

Tools and Methodology

The project objective is to create the program that be able to recognize exercise type and count how much time each exercise has been done. The data set have been created in 3 different image classes which are set, knee touch and squats down as shown in the figure 3 - 5. To recognize the exercise motion, the program recognizes the difference between the two frames gesture if the frame prediction have change from set to squats down and come back to set position. Then the program will count that user have done 1 cycle of squats. The program can be represented as a state machine as shown in figure 6. It shows that if user has done the gesture correctly, whether it is a squat or knee touch, the program will move to the next state, but the output will still be zero (count not yet started). But when the user returns to the initial position, then the program will add one to count into the program data while running.



Figure 3. Set



Figure 4. Knee touch



Figure 5. Squats down

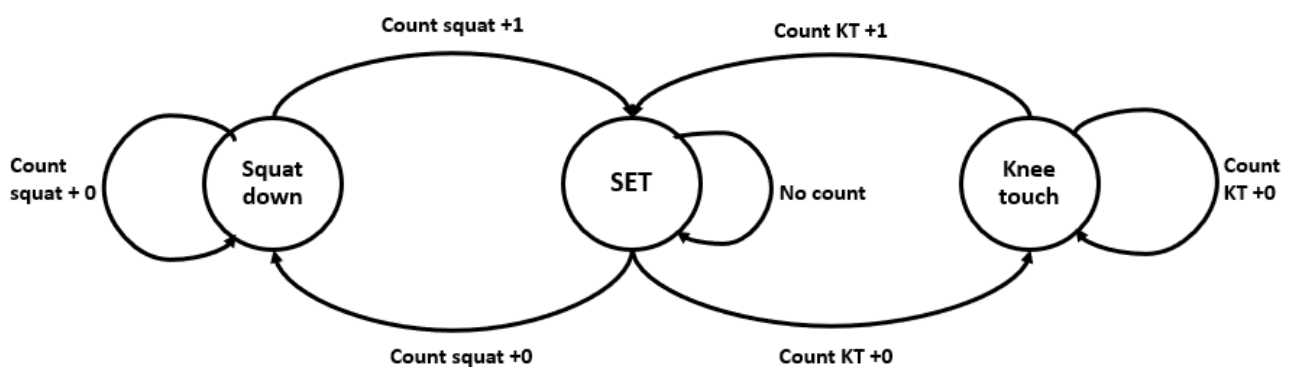


Figure 6. State machine representing the process of exercise counting

Dataset collection

In this project, the dataset has been collected to three images classes namely set, knee touch and squats down for classification process of the human gesture as shown in the figure 3 - 5. To avoid the bias and error on training the model for classifier, we clean the dataset that have invalid posing and miss pose more than 2 pose joint point as shown in figure 6 - 8. To add a feature to the data set, data augmented have been used to increase the amount of data by adding features including shifting the image to the left and right. This creates new and different images from the existing image data set that represents a comprehensive set of possible images. Deep learning Convolutional Neural Networks (CNN) . need a huge number of images for the model to be trained effectively. This helps to increase the performance of the model by generalizing efficiently and thereby reducing overfitting.

Furthermore, the process of extracting images before training the model by checking all of the key joints are complete or can be detected precisely, for example in Figure 6. The exact position of the arm could not be determined. Figure 7, 8 algorithm detects the wrong leg position. Thus, images like this are considered bad images that will not be used to train model further.



Figure 8. Invalid set pose

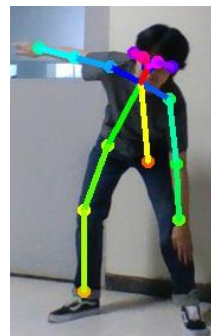


Figure 9. Invalid knee touch pose

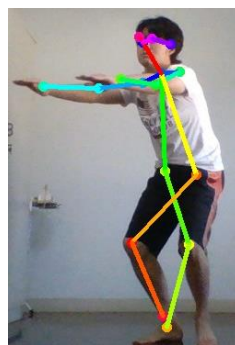


Figure 10. Invalid squat down pose

Model for pose estimation

From table 1 shows the performance of all 4 models posing and result of real time FPS working with GPU. Then, the accuracy of the number of fault tracking or missing key point by classification accuracy which is one metric for evaluating classification models was calculated. Informally, accuracy is the fraction of predictions model getting correct result. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Table 1. performance of each pose estimation model

Model	Real time FPS	Model size	Computation time	Key Points	Accuracy
CMU	3 – 9 FPS	204394 KB	0.091 s	18/18	100 %
Thin MobileNet	20 – 40 FPS	7622 KB	0.044 s	17/18	94.44 %
Small MobileNet v2	25 – 40 FPS	2047 KB	0.028 s	14/18	87.78 %
Large MobileNet v2	20 – 36 FPS	8882 KB	0.051 s	15/18	83.33 %

The outcome that have shown in table 1. show that the Thin MobileNet provided the most satisfactory results namely data from this model is stable, correct and Thin MobileNet is light weight model which will has an enormous effect to the result of this project. Therefore, Thin MobileNet has been selected for further classification process of this project due to the fact that this model was showing the best performance for real time pose-estimation when compared with another one.

Classifying Gesture Algorithm

Identification exercise gesture was brought about by TF-pose-estimation. Firstly, gathering information for processing with Thin MobileNet. In this process, the output is produced in 18 different points of the body as shown in Figure 1. After that, the position data of each point is calculated with the multilayer perceptron (MLP) to distinguish which gesture is being performed as in Figure 9 (set, knee touch, squat down).

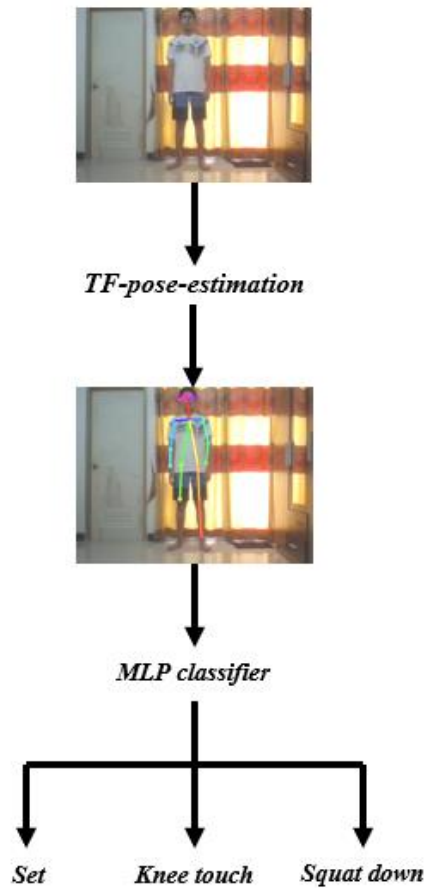


Figure 11. Algorithm for classifying exercise gesture

Results and Discussion

Test run




The method used in this project to evaluate the performance of the model had been tested by taking 100 time per class and calculate the accuracy of the model. The result is shown.

In table 2, the statistic information is provided from test running each exercise pose and the program has shown the confidence rate as indicates in Figure 10. If confidence rate below than 95% the program will not predict anything and will provide NAN class.

```
Check: set | Confident: 0.9520973063609325
Check: nan | Confident: 0.8546965180865204
Check: set | Confident: 0.9559145261440982
Check: nan | Confident: 0.9404444354164119
Check: nan | Confident: 0.7030615354580243
Check: knee_touch | Confident: 0.9908327703965474
Check: knee_touch | Confident: 0.9709557672120056
Check: knee_touch | Confident: 0.9998455613982757
Check: knee_touch | Confident: 0.9999856479321387
Check: knee_touch | Confident: 0.9999905616765599
```

Figure 12. shows the program while running

Table 2. shows the confidence rate with correctness answer

Pose	Picture	Recognize Class	Confidence
Set		set	99.99%
Knee touch		knee_touch	97.41%
Squat down		squat_down	98.27%

Real time estimation

From the trained dataset of this program which came from a limited dataset, is lack of diversity of information. Therefore, only results in certain conditions that model can predict correctly with accurate and precise is used. That also includes a matter of term correct posture including angles and degrees of pose and minor details such as the color of the clothes also considered to create confidence that is below 95%. As show in Table 3 – 4 and Figure 13 – 14. Notice that sometime the model detected fault gesture due to the error of the posing from TF-pose-estimation. This might be another error to the prediction of the model.

Table 3. shows the result which following the condition as well as trainer dataset

Pose	Count	Total	Accuracy
Knee touch	98	100	98%
Squat down	91	100	91%
Total	189	100	94.5%

Table 4. shows the result which does not meet the condition

Pose	Count	Total	Accuracy
Knee touch	87	100	87%
Squat down	17	100	17%
Total	104	200	52%

From the Table 4. If user does not follow the conditions such as wrong pose of distance between camera and human too short or too far, the model will predict wrongly which squat down would consider as knee touch pose etc.

Conclusions

The application of exercise pattern recognition approach is used for classifying the input images by Multilayer Perceptron method. Training set feature is generated by exercise gesture from trainer who provided advance performance images for further training into TF pose estimation which consists of set, knee touch and squat down. Simulation result shows that high accuracy classifiers with less fault dismissal rate are realizable with classifier algorithm with some of the condition due to the low variety of the data make model over-fitting.

Appendix A:

Tools specification

CPU

- AMD Ryzen 9 4900H
- 8 cores 16 threads

TensorFlow

- TensorFlow Version 2.1.0
- TensorFlow-gpu Version 2.1.0

GPU

- NVIDIA Geforce RTX2060 (GDDR6 8GB)

CUDA

- CUDA version 10.1
- Cudatoolkit 10.1.243

CUDNN

- version 7.6.5

OpenCV

- version 4.4.0.44

Scikit-learn

- version 0.23.2

Pandas

- version 1.1.4







Numpy

- version 1.19.2

Appendix B:

Original and augmented data comparison

Table 5. shows the sample comparison between original image and augmented image

Pose	Original image	Augmented image
Set		
Knee touch		
Squat down		

Reference

- [1] Artyom Kulakov (2010 April). *“How I created the Workout Movement Counting App using Deep Learning and Optical Flow Algorithm”* [Online] Available:<https://towardsdatascience.com/how-i-created-the-workout-movement-counting-app-using-deep-learning-and-optical-flow-89f9d2e087ac> [Accessed Nov 27, 2020]
- [2] Yoli Shavin (2009 July). *“Introduction to Camera Pose Estimation with Deep Learning”* [Online]. Available:https://www.researchgate.net/publication/334362201_Introduction_to_Camera_Pose_Estimation_with_Deep_Learning [Accessed: Nov. 27, 2020].
- [3] Andrew G. Howard (2017 April). *“MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”*. Cornell University.
- [4] Angelo Yao, Juergen Gall, Gabriele Fanelli and Luc Van Gool. *“Does Human Action Recognition Benefit from Pose Estimation?”* Computer Vision Laboratory, ETH Zurich, Switzerland, 2010.
- [5] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *“TensorFlow: Large-scale machine learning on heterogeneous systems”* Software available from tensorflow.org, 2015.
- [6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh *“OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”*. Student Member, IEEE. [cs.CV] (2019 May)

[7] Mykhaylo Andriluka. (2015) “*2D Human Pose Estimation: New Benchmark and State of the Art Analysis*”. Stanford University.

[8] Crescenzo Gallo “*Artificial Neural Networks: tutorial*”, Università degli studi di Foggia, 2015 January.

[9] John McGonagle, George Shaikouski and Christopher Williams “*Backpropagation*” [Online] Available: <https://brilliant.org/wiki/backpropagation/#:~:text=Backpropagation%2C%20short%20for%20%22backward%20propagation,to%20the%20neural%20network's%20weights.> [Accessed Nov 27, 2020].

[10] Kalyani & K.Shanti Swarup. “*Study of Neural Network Models for Security Assessment in Power System*” Indian Institute of Technology Madras, 2009 November