Human Gym Activity Recognition using RGB Camera

Preetham Ganesh
preetham.ganesh@mavs.uta.edu
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX, USA

Chinmaya Venkatesh
chinmaya.venkatesh@mavs.uta.edu
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX, USA

Reza Etemadi Idgahi
reza.etemadi.idgahi@mavs.uta.edu
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX, USA

ABSTRACT

Human activity recognition is one of the most researched topics in computer science. It is a powerful tool that can be used aiding medical systems, assisted living in a smart home and many more areas. In this project, gym activities such as push-up, squat, plank, forward lunges and sit-up are chosen as the list of activities, and an RGB camera was used to record the activities performed by the four participants. The recorded videos were fed into OpenPose for skeleton point extraction, and the extracted skeleton points are fed into classification algorithms such as Naive Bayes, Support Vector Machines and Decision Tree Classifier for classification after the pre-processing the raw skeleton points. The developed models are evaluated using performance measures such as Accuracy and Balanced Accuracy. Also, the accuracy of the workout performed user was calculated using the regression performance metric called as Explained Variance Score for each skeleton point. The output was displayed in an android application with an attractive and usable user interface.

KEYWORDS

Human Activity Recognition, Open Pose, Classification

ACM Reference Format:
Preetham Ganesh, Chinmaya Venkatesh, and Reza Etemadi Idgahi. 2018. Human Gym Activity Recognition using RGB Camera. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Human Activity Recognition (HAR) is currently one of the highest researched topics in the field of image processing [1]. It mainly focuses on recognising an action and aim of the agent(s) who performs it using the previous behavior of the agent and environmental conditions. Different sensors can be used to obtain the data of activity such as RGB camera, inertial sensors, depth sensor and many more. HAR can be classified into the following types: Sensor-based single user activity recognition [2], Sensor-based multi-user activity recognition [3] and Sensor-based group activity. HAR has various applications such as assisted living in a smart home, health care monitoring systems, wildlife observation and many more [4].

The topic discussed in this report belongs to the second category mentioned above, and the list of activity used is a more specific one, i.e. Human Gym Activity Recognition. The activity monitoring can also be used to improve the health of a human being; in other words, it can change the lifestyle of a person for good. Healthy habits include eating well (proper food such as grains, fat-free meat, dairy products with low fat), sleeping well and exercising well, in other words avoiding a sedentary lifestyle. A sedentary lifestyle of a person includes sitting and lying down while doing activities such as reading books and watching television and many more [5]. A healthy lifestyle can keep a person's weight under check, which can fight against diseases such as diabetes, depression, and many more. But changing the lifestyle of a person has always been a challenging task due to a few reasons. Firstly, many do not know which kind of gym activities has to be performed; secondly, the said person may not have time to do the workout as per their schedule; thirdly, the person may not like being monitored by another person to check the accuracy of the exercise. The proposed work is an attempt at providing solutions for the problems mentioned above which is explained in Section 4.

The composition of the proposed work is as follows: Section 2 discusses about the related works in the field; Section 3 talks about the machine learning methods and the performance measures used for the development of the prediction models; Section 4 talks about the system architecture used for the proposed system; Section 5 discusses about the experimental results the were received on using the proposed method; and Section 6 discusses about the conclusion we arrived at on using the proposed method.

2 RELATED WORKS

This section discusses in detail the various works that have been done in recent times in the field. It also talks about the different prediction methods that were used by the authors for prediction purposes, even the error measures used for testing the performance of the developed models.

The authors in [6] performed human activity recognition, where they chose ten domestic (everyday) activities and eleven sports activities for the same. For recording the data, the authors used two wearable sensor devices which consist of Micro-controller, Triaxial Accelerometer, Triaxial Gyroscope, RF Wireless Transmission Module and Power Supply Circuit. The paper consists of various algorithms for modules such as motion signal acquisition, signal

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

pre-processing, dynamic human motion detection, signal normalization, feature extraction, feature normalization, feature reduction and activity recognition. Principal Component Analysis (PCA) was used to reduce the complexity, and 10-fold cross-validation was used to validate the effectiveness of the developed model which tested with the help of performance measures such as Correct Classification Rate, Accuracy, Specificity and Sensitivity.

O. M. Prabowo, K. Mutijarsa, S. H. Supangkat in [7] tried to apply missing data handling using machine learning on human activity recognition where actions recorded using an RGB camera. The RGB camera used was on Xiaomi Redmi 2, which comes with accelerometer, gyroscope, digital compass, microphone and Geographical Positioning System (GPS). The activities used for testing are walking, and sitting and algorithms used for classification are Bayesian Network, Multiple Layer Perceptron (MLP), C4.5 Decision Tree and k-Nearest Neighbours (kNN). The missing data percentages used for testing are 5%, 10%, 20%, 30% and 40%. Similar to the paper mentioned above the authors in this paper used 10-fold cross-validation for the validation of the models. Using accuracy as the performance measures, the authors concluded that kNN outperformed the other models.

The authors in [8] used LIBSVM (a library on Support Vector Machines) on the accessible datasets i3DPosts and IXMAS (INRIA Xmas Motion Acquisition Sequences). The datasets consisted of 13 daily activities such as check watch, cross arms, and many more. For evaluating the developed model, the authors used validation methods such as 5-fold cross-validation and leave one out cross-validation. Using accuracy as the performance measure, the authors concluded that the proposed method performed better the current approaches in both the datasets.

K. Xu et al. in [9] developed a novel method in developing a HAR model, which is not affected by the parts such as hair, face, shadow, clothes texture or the shake of the camera. For this purpose, the authors used the HOG human detector and Contour detector for detecting human body edges. The authors implemented the method mentioned above on Weizmann dataset and KTH dataset, and the exposed edges were passed as input into the SVM classification algorithm. The developed model produced an accuracy of 99.1 on Weizmann dataset and 95.8 on KTH dataset.

M. Babiker et al. in [10] developed a HAR model which extracts features from the images in the database using digital image processing techniques such as background subtraction, binarization, and morphological operation. The extracted features were sent into a multi-layer feed-forward perception network to classify the activities. Accuracy was used as the performance measure to evaluate the developed model, and the model performed well in training, validation and testing stages.

The authors in [11] developed a single dimensional Convolutional Neural Network (CNN) to classify actions such walking, running, and staying still, where the data was recorded using the accelerometer in mobile phone. The authors also used Random Forest Classifier (RFC) for comparison, where the performance measure used for evaluating the model is accuracy. It was concluded by the authors that the single dimensional CNN outperformed the RFC. M. C. Sorkun et al. in [12] developed a HAR by collecting data from 15 participants for six different activities, and the collected data was fed into the feature extraction model and the extracted features are fed into five classification algorithms such as SVM, Multiple Layer Perceptron, K-Nearest Neighbors (KNN), RFC and Naive Bayes. The developed models are compared using performance measures such as accuracy.

H. Huang et al. in [13] a HAR model by using the triboelectric sensor for collecting the motion signals of activities such as sitting and standing, walking, climbing upstairs, downstairs, and running and the recorded data were classified using KNN clustering algorithm and accuracy was used to evaluate the performance of the developed model. It was concluded that the developed model produced an accuracy of 80%. The authors in [14] used pyroelectric infrared (PIR) sensor arrays, feature extractor, and a classifier to develop a HAR model that is capable of recognizing a list of 6 activities. The collected data were converted into binary numbers as inorder to preserve the geometry and motion information of the target. The authors concluded that the developed model is capable of recognizing the activities with just a few bits of data.

M. Ehatisham-Ul-Haq et al. in [15] used the dataset UTD-MHAD (Univerisity of Texas at Dallas - Mult-Human Activity Dataset) for developing HAR which combines the data from different sensors such RGB camera, depth sensor and inertial sensor. Skeleton points in the RGB videos are obtained by using HOG descriptor and the data from the other sensors are pre-processed and fed into classification algorithms in different combinations. The classification algorithms used are SVM and KNN, and the accuracy measures used for evaluating the developed models are accuracy, precision, recall and f-measure. It was concluded by the authors that on combining all the data from different sensors and applying KNN on the combined data produced the best results.

Among all the papers mentioned above, not many articles used gym activities as the set of activities to be recognized, and the papers calculate the accuracy of the prediction but the accuracy of the actions performed by the user is not calculated. The work proposed in this project uses a list of gym activities such as push-up, squat, plank, forward lunges and sit-up as the activities to be recognized by the model, and the accuracy of the action performed by the user is calculated and displayed in an android application.

3 METHODOLOGY

This section discusses in detail the classification algorithms used for predicting the workouts, error measures used for testing the performance of the developed models.

3.1 Classification Algorithms

3.1.1 Support Vector Machine. It is a supervised learning method which can be used in both regression and classification analysis [16]. The technique represents the given points in space and divides the points based on the categories and uses the boundaries to separate them. When new points are entered, the points are plotted into the space to classify the points based on the computed border.

3.1.2 Naive Bayes Classifier. It is a probabilistic classification algorithm majorly based on the mathematical concept Bayes Theorem, and it works on highly independent feature mechanism [17]. The classifiers under this mechanism are majorly expandable and need a vast number of parameters which are linear in nature.

3.1.3 Decision Tree Classifier. It is a statistical analysis method predominantly used in machine learning and natural language processing [18]. A typical decision tree consists of three nodes: Decision nodes, Chance nodes and End nodes [19].

3.2 Performance Measures

3.2.1 Confusion Matrix. Also commonly known as Error Matrix, is used for visualizing the performance of a classification model where the visualization shows the performance of the model for each label in the class [20].

3.2.2 Accuracy. It is a metric for testing the performance of the classification model developed. It is formally defined as the fraction of the predictions that are correct by the total number of predictions. The equation in 1 shows the formula for accuracy used in this project.

$$Accuracy = \frac{TL_1 + TL_2 + TL_3 + TL_4 + TL_5}{N} \tag{1}$$

Where TL_i represents the total number of accurate predictions for the corresponding label L_i and N represents the total number of the predictions by the model.

3.2.3 Balanced Accuracy. It is computed as the mean of the percentage of the correct instances for each class individually. It is an excellent measure to check the performance of the classification model when the number of rows per class is not the same or do not fall in the same range.

3.2.4 Explained Variance Score (EVS). It is used to measure the discrepancy between a model and actual data. It is the part of the model's total variance that is explained by factors that are actually present and isn't due to error variance. The equation is shown in (2).

$$EVS = 1 - \frac{Variance(Y_t - Y_p)}{Variance(Y_t)} \tag{2}$$

4 PROCESS FLOW

The videos used for training the machine learning model was recorded using the RGB camera in a Google Pixel 3a mobile phone as shown in Fig. 1. The videos were recorded in a frame size of 1920 x 1080 with 30 frames per second. The recorded videos contain five gym activities namely: push-up, squats, plank, forward lunges and sit-up. In order to bring in a variety, 4 subjects were asked to perform the workout in front of the camera where the subjects were informed about the details of the project and were asked to sign a consent form explaining the said details. Each subject had to perform 15 repetitions of push-ups, 15 repetitions of squats, 1 repetition of plank for 30 seconds, 15 repetitions of forward lunges and 15 repetitions of sit-ups. The examples of frame representation for each of the activities are shown in Fig. 2. The recorded videos were saved in a format similar to the work in [21], i.e. 'a1_s1_r1.mp4', where 'a' stands for activity, 's' stands for subject and 'r' stands for repetition.

Each of these videos are processed using the trained convolutional neural network (OpenPose) developed in [22]. The video is loaded and is saved as a list of frames, where each frame is resized from 1920x1080 resolution to 640x480 resolution for easier execution. The OpenPose function uses the resized frames to generate the skeleton points, where function was executed on a HP Envy 13 T system with configuration of (CPU: Intel 8^{th} Generation (8565U) Core i7 processor, GPU: GTX MX 250). The execution took a time of 22 seconds to process a frame in a video, where a total of 18 skeleton points are generated for each frame (if recognized) which include nose, neck, eyes, ears, shoulders, elbows, wrists, hips, knees and ankles [23]. Table 1 shows the list of key points generated by Caffe for a frame. The generated key points are saved as a text file for further processing and an example of skeleton point in a text file for one frame is shown in Fig. 3.

Skeleton Point No.	Body Part
0	Nose
1	Neck
2	Right Shoulder
3	Right Elbow
4	Right Wrist
5	Left Shoulder
6	Left Elbow
7	Left Wrist
8	Right Hip
9	Right Knee
10	Right Ankle
11	Left Hip
12	Left Knee
13	Left Ankle
14	Right Eye
15	Left Eye
16	Right Ear
17	Left Ear

Table 1: List of skeleton points and their corresponding body parts

The text file generated is converted into a Comma Separated File (CSV), which has 55 columns. The first 18 columns represent the x values of skeleton points in the frame, next 18 columns represent the y values of skeleton points in the frame, following 18 columns represent the confidence of the OpenPose in generating the skeleton point and the last column represents the frame number the skeleton points belongs to. The generated CSV files have a lot of missing values as only the right-hand side of the subject is visible to the camera; hence, columns pertaining to skeleton points left-hand side of the body are dropped. The balance missing values in the data frame are imputed the using frame numbers, i.e. the frame number of the missing point is selected and the values belonging to the corresponding columns from the subject's other videos for

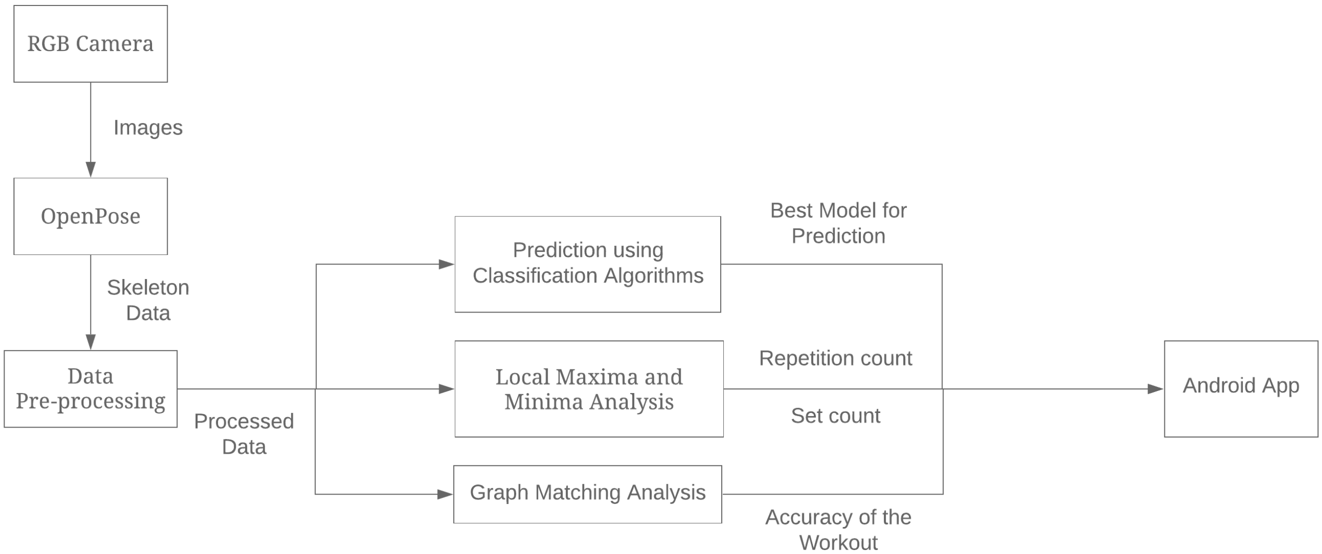
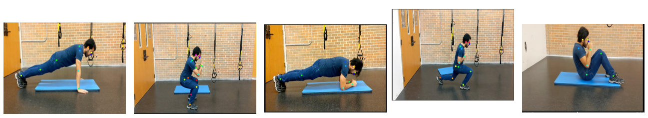


Figure 1: System Architecture



(a) Push-up (b) Squat (c) Plank (d) Forward Lunge (e) Situp

Figure 2: Examples of frame representation for each of the activities

[[(457, 265, 0.22543591260910034, 0)], [(413, 209, 0.12301988899707794, 1)], [(412, 203, 0.18981756269931793, 2)], [(381, 253, 0.223374143242836, 3)], [(389, 336, 0.20884007215499878, 4)], [], [], [], [], [(202, 266, 0.17279966175556183, 5)], [(98, 266, 0.17263711988925934, 6)], [], [], [], [(462, 254, 0.23444101214408875, 7)], [], [(451, 230, 0.23091995716094971, 8)], []]

Figure 3: Sample of skeleton keypoints for a frame in a video

the same activity and the same frame number. If the value is not available, then the mean value of the column is chosen as the value.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

This section discusses in detail about the results obtained on implementing classification algorithms on the final CSV file generated in Section 4, calculating the number of iterations a user has performed and computing the accuracy of the activity performed by the user.

5.1 Correlation between the Skeleton Points and the Workout

The final CSV file was used for this purpose, where correlation was found between x values, y values and the confidence values of the skeleton points, and the activity. The Pearson Correlation Coefficient was used for this purpose and the formula for which is given in the equation in (3).

$$\rho_{\mathrm{x,\,y}} = \frac{\sum(X_{\mathrm{i}} - X_{\mathrm{m}})(Y_{\mathrm{i}} - Y_{\mathrm{m}})}{\sqrt{\sum(X_{\mathrm{i}} - X_{\mathrm{m}})^2 \sum(Y_{\mathrm{i}} - Y_{\mathrm{m}})^2}} \tag{3}$$

Where X and Y are the continuous variables. X_i and Y_i represents the i^{th} element in the vectors and X_m and Y_m are the mean values of the corresponding vectors. Heat map is used for visualizing the generated correlation matrix, where the image in Fig. 4 shows the correlation between the x values of the selected skeleton points and the activity. Similarly, Fig. 5 shows the same for y values of the skeleton points and Fig. 6 shows the same for confidence values of the skeleton points.

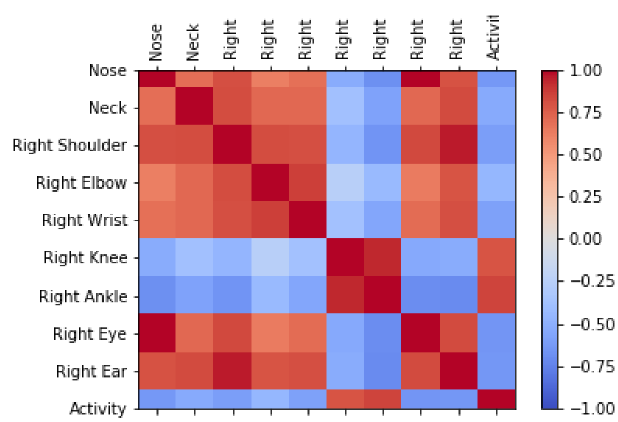


Figure 4: Correlation between the x values of the selected skeleton points and the activity

It can be observed from the image in Fig. 4 that the skeleton points belonging to the body parts Nose, Neck, Right Shoulder,

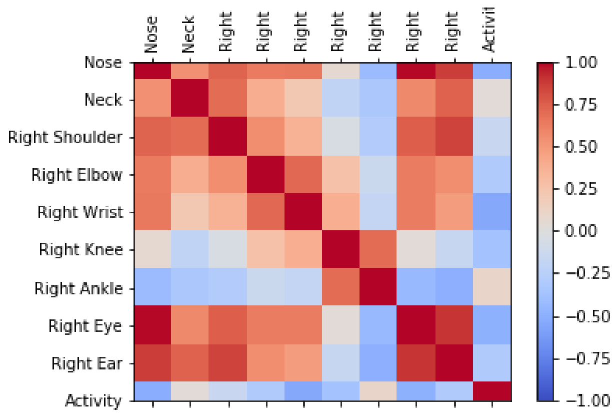


Figure 5: Correlation between the y values of the selected skeleton points and the activity

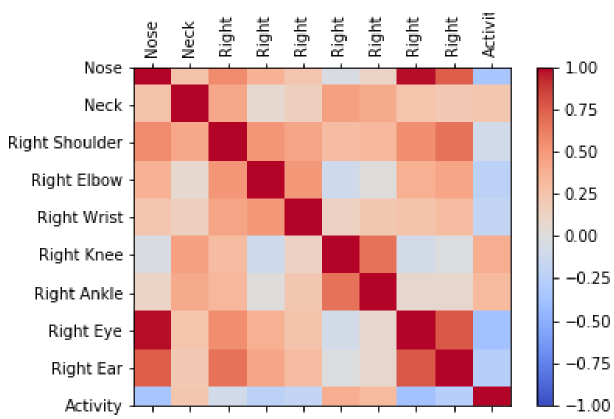


Figure 6: Correlation between the c values of the selected skeleton points and the activity

Right Elbow, Right Wrist, Right Eye and Right Ear are negatively correlated with the activity and the other attributes are positively correlated. It is because of the location of the body parts. A similar scenario can be observed in the image in Fig. 5, where the intensity or the value of correlation differs, but the observation is the same.

Similarly, it can be observed from the image in Fig. 6 that the confidence values for all the skeleton points are all positively correlated. It can also be commonly observed that in all the images mentioned above that correlation values are approximately higher than the absolute value of 0.25.

5.2 Workout Classification

To perform the classification efficiently, all the columns in the CSV file was normalized using Min-Max normalization as it will help reducing the time taken to train the model. The formula used for normalization is shown in (4).

$$X'_{\mathrm{i}} = \frac{X_{\mathrm{i}} - X_{\min}}{X_{\max} - X_{\min}} \quad (4)$$

Where X_i is the i^{th} element in the feature, X_{min} is the minimum value of the feature, X_{max} is the maximum value of the feature and X'_i is the normalised value of the i^{th} element in the feature. The normalized columns are added into a dictionary and converted into a data frame. The normalized data frame was fed into the classification algorithms mentioned in Section **??**. To evaluate the performance of the trained models, Repeated K-Fold cross validation was used, where the number of repeats is 10 and the number of folds is 10. The performance values on using the algorithms on the normalized CSV file is shown in Table 2.

It can be observed from the table that all the classification models have high accuracy and high balanced accuracy in both the training and validation set. It should be noted that the results shown above belong to the basic version of the model, i.e. the hyper-parameters were not changed. It can also be observed that the Decision Tree Classifier model outperforms the other models in both training and validation and hence is chosen for final prediction.

5.3 Estimation of Accuracy of the Workout

For each of the activity, a reference file is chosen which is used for computing the accuracy of the workout performed by the user. But, the length of the files can vary, hence the user file length is changed, i.e. if length is more it is decreased and if the length is less it increased by filling the new values with the mean value of the corresponding column. The accuracy is computed for every skeleton point with the help of Explained Variance Score where true value is the reference value and predicted value is the user's value. Since, the scale of these columns can change as the user can perform the workout at any depth from the camera, the columns are normalized using Min-Max Normalization. The EVS values are calculated separately for x and y dimensions and the mean value of these dimensions are taken as the accuracy for a particular skeleton point. The accuracy for each skeleton point are split into 6 categories which are shown in Table 3.

5.4 Android Application

The workout patterns of the user are monitored using an android application which consists of four pages namely Home page, Setting page, Report page and Workout Detail page. Django server was used to integrate and run the web camera and received the processed data and results. The android app pings the Django server for every 10 seconds to check whether the required results are available to populate the Android Application. Fig. 7a represents a pie chart with the number of repetitions for each workout. Fig. 7b depicts the home page where the user can start and stop the time of his workout Fig. 8a represents the Settings page where the user sets the to connect to the Django server Fig. 8b represents the Detailed workout page along with the accuracy of each posture.

6 CONCLUSION

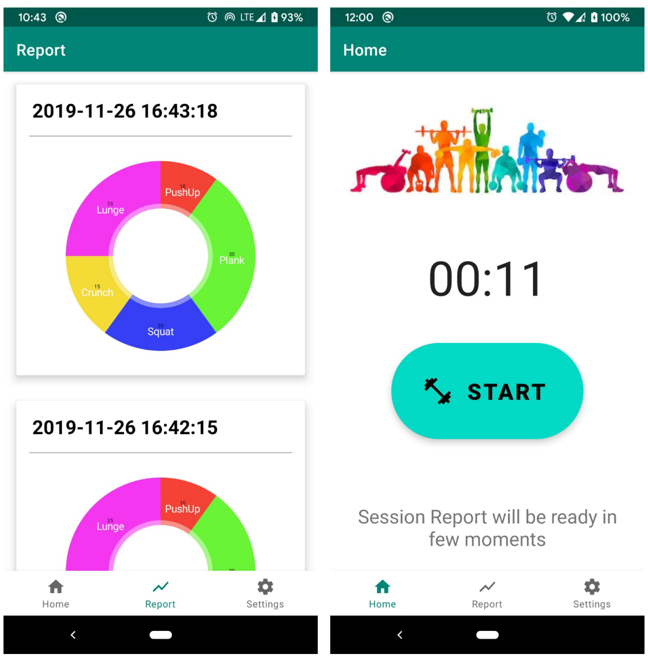
In this project gym activities such as the push-up, squat, plank, forward lunge and sit-up are used as the list of activities for recognition. OpenPose was used for extracting the skeleton point from

Model Name	Accuracy		Balanced Accuracy	
	Train Set	Validation Set	Train Set	Validation Set
GNB	0.9648	0.9647	0.9748	0.9747
SVM	0.9764	0.9763	0.9812	0.9812
DTC	**1.0**	**0.9983**	**1.0**	**0.9985**

Table 2: Performance of the classification algorithms on the CSV file

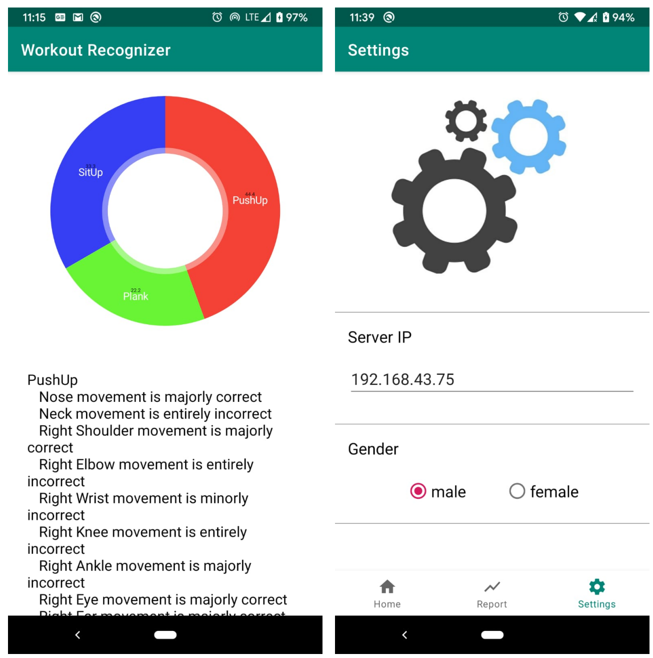
S.No	Accuracy Range	Definition
1	Accuracy <0	Movement is entirely incorrect
2	Accuracy >0 and Accuracy <0.2	Movement is majorly incorrect
3	Accuracy >0.2 and Accuracy <0.4	Movement is minorly incorrect
2	Accuracy >0.4 and Accuracy <0.6	Movement is minorly correct
2	Accuracy >0.6 and Accuracy <0.8	Movement is majorly correct
2	Accuracy >0.8 and Accuracy <1.0	Movement is entirely incorrect

Table 3: Categories of Accuracy Spliting



(a) Report Page (b) Home Page

Figure 7: Screenshot of pages on Android Application



(a) Detailed Workout Page (b) Settings Page

Figure 8: Screenshot of pages on Android Application

the videos, and the extracted skeleton points were pre-processed and fed into classification algorithms, namely Naive Bayes, Support Vector Machine and Decision Tree Classifier. It was observed that the basic version of the classification models produced outstanding results. Using accuracy and balanced accuracy as the performance measures, it was concluded that the Decision Tree Classifier outperformed the other classification models. Also, the accuracy in the movement of the joints in the workout performed by the user was calculated with the help of the regression metric explained variance score. The detailed report was presented to the user in an android application with an attractive user interface.

ACKNOWLEDGMENT

We would like to thank Dr Fillia Makedon (Professor, Department of Computer Science and Engineering, The University of Texas at Arlington), Dr. Maria Kyrarini (Postdoctoral Researcher, Department of Computer Science and Engineering, The University of Texas at Arlington), Mr. Maher Abujelala (Ph.D Student, Department of Computer Science and Engineering, The University of Texas at

Arlington), Mr. Ashwin Rameshbabu (Ph.D Student, Department of Computer Science and Engineering, The University of Texas at Arlington and Mr. Akilesh Rajavenkatanarayan (Ph.D Student, Department of Computer Science and Engineering, The University of Texas at Arlington) for their significant critiques during the course of the project.

REFERENCES

[1] M. Ehatisham-Ul-Haq, A. Javed, M. A. Azam, H. M. A. Malik, A. Irtaza, I. H. Lee, and M. T. Mahmood. Robust human activity recognition using multimodal feature-level fusion. *IEEE Access*, 7:60736–60751, 2019.

[2] T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, B. Harrison, B. Hemingway, J. Hightower, P. P. Klasnja, K. Koscher, A. LaMarca, J. A. Landay, L. LeGrand, J. Lester, A. Rahimi, A. Rea, and D. Wyatt. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7(2):32–41, April 2008.

[3] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, January 1992.

[4] Suneth Ranasinghe, Fadi Al Machot, and Heinrich C Mayr. A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, 12(8):1550147716665520, 2016.

[5] Physical Activity Guidelines Advisory Committee Scientific Report, 2018.

[6] Y. Hsu, S. Yang, H. Chang, and H. Lai. Human daily and sport activity recognition using a wearable inertial sensor network. *IEEE Access*, 6:31715–31728, 2018.

[7] Okyza M Prabowo, Kusprasapta Mutijarsa, and Suhono Harso Supangkat. Missing data handling using machine learning for human activity recognition on mobile device. In *2016 International Conference on ICT For Smart Society (ICISS)*, pages 59–62. IEEE, 2016.

[8] N. Käse, M. Babaee, and G. Rigoll. Multi-view human activity recognition using motion frequency. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3963–3967, Sep. 2017.

[9] K. Xu, X. Jiang, and T. Sun. Human activity recognition based on pose points selection. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2930–2834, Sep. 2015.

[10] M. Babiker, O. O. Khalifa, K. K. Htike, A. Hassan, and M. Zaharadeen. Automated daily human activity recognition for video surveillance using neural network. In *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, pages 1–5, Nov 2017.

[11] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 131–134, Feb 2017.

[12] M. C. Sorkun, A. E. Danişman, and Ö. D. İncel. Human activity recognition with mobile phone sensors: Impact of sensors and window size. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, May 2018.

[13] H. Huang, X. Li, and Y. Sun. A triboelectric motion sensor in wearable body sensor network for human activity recognition. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4889–4892, Aug 2016.

[14] G. Liu, J. Liang, G. Lan, Q. Hao, and M. Chen. Convolution neutral network enhanced binary sensor network for human activity recognition. In *2016 IEEE SENSORS*, pages 1–3, Oct 2016.

[15] M. Ehatisham-Ul-Haq, A. Javed, M. A. Azam, H. M. A. Malik, A. Irtaza, I. H. Lee, and M. T. Mahmood. Robust human activity recognition using multimodal feature-level fusion. *IEEE Access*, 7:60736–60751, 2019.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[17] David J. Hand and Keming Yu. Idiot's bayes: Not so stupid after all? *International Statistical Review / Revue Internationale de Statistique*, 69(3):385–398, 2001.

[18] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986.

[19] Bogumił Kamiński, Michał Jakubczyk, and Przemysław Szufel. A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research*, 26(1):135–159, Mar 2018.

[20] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89, 1997.

[21] C. Chen, R. Jafari, and N. Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 168–172, Sep. 2015.

[22] S. Qiao, Y. Wang, and J. Li. Real-time human gesture grading based on openpose. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–6, Oct 2017.

[23] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.