Journal of Engineering Sciences
Vol 12, Issue 06, JUNE /2021
ISSN NO:0377-9254

AN EFFECTIVE HUMAN ACTIVITY RECOGNITION USING CNN WITH AUTOENCODER BASED ON SMARTPHONE DATA

[1]Mrs.Sri Chintha Jyothi [2]Mr.L.V. Kiran [3]Mr.K. Praveen Kumar

[1]PG Student, [2]Assistant professor [3]Assistant professor

[123] Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh

[1]Srichintha.jyothi@ gmail.com [2]lvkiran@giet.ac.in [3]praveenkumar@giet.ac.in

ABSTRACT:

The use of smartphone sensors to predict current user activity is a very useful feature for most mobile applications. However, for low-cost sensors and real-life applications, this task can easily become a problem. The main question is how to reduce the number of Human Activity Recognition (HAR) errors in everyday smartphone use. The system investigates the effects of smartphone diversity on next-generation models and finds solutions such as trend-independent preprocessing transformation techniques to reduce it. In addition, in the current literature, we investigate the advantages of combining the features extracted from a convolutional neural network (CNN) based car encoder compared to the most well-known model in the current literature. Next, we test our architecture first with a different data set that exceeds the previous results, then with a new dataset of real-world use cases in this case for the best results.

Key Terms — Human Activity Recognition, Convolutional Neural Networks, Autoencoders

1.INTRODUCTION:

It is important to identify day-to-day human activities for many programs, from health monitoring to safety monitoring, fitness monitoring and user adaptation systems. With the development of low-end sensors in smartphones and mobile devices, developing mobile applications that can monitor user activity "in nature" is a major challenge. As mentioned, consumers and smartphones have many variables. Users have different differences in demographic terms (age, height, weight, etc.) and different functions in style. Instead, they use different operating systems, hardware, and detection capabilities. We start with the proposed model and show the controlled environmental impact of the classification. The break between the device (and the sensor) does not cause data corruption and an unnatural environment. For this reason, we accept provider data sets highlighting the diversity of devices. We also examined the effectiveness of the manual design and statistical functions which are often inadequate in real life due to heterogeneous situations. In addition, we considered the effect of orientation-independent transformation as a preprocessor, which should render the data independent of sensor location and orientation. we aim to discuss the real-life disadvantages of diversity using the older generation model and then create a new learning framework to address the disadvantages of HAR in this case. Our framework is built on the CNN architecture combined with the automatic portable encryption features of mobile devices. Finally, we will compare our results with pioneering work.

The mode of transportation used by most smartphone users plays an imperative role in the amount of contextual information used by the user's mobility while travelling. Using multimodal data, apps can customize services to make tracking people smarter, easier and smarter. In recent years, numerous studies have identified ways to transmit GPS and motion data from sensors mounted on smartphones. Most of these studies used classical machine learning and

deep learning models to define behavioral modes. The outcomes display that it is difficult to distinguish vehicles from the same category with the motion sensor approach.

This device is also equipped with an accelerometer and gyroscope. Accelerators are standard in almost all smartphone manufacturers. As the name implies, the accelerometer measures changes in speed. not speed. Acceleration data can be managed to sense sudden deviations in motion. Another standard smartphone sensor is a gyroscope, which uses gravity to measure orientation. The signal received by the gyro can be managed to control the location and positioning of the device. Because there are significant property differences between the data obtained from these sensors, many properties can be generated from the sensor data to determine the activity of the person using the device. In this study, the data set consisted of signals from smartphone accelerometers and gyroscopes launched by different male and female volunteers while performing various covert tasks using CNN's machine learning approach. The effectiveness of various approaches is analyzed and compared in terms of accuracy and efficiency.

2.PROPOSED SYSTEM

If we are actually used, we need to take into account that smartphones can be adapted and targeted to the human body in various ways. For example, a smartphone can be in a trouser pocket (front and back), or it can be in a purse or bag in different directions. These variables have a significant impact on the prediction of HAR predictions, especially when the model is trained to collect data that includes measures of performance along with a set of directions and orientations, usually for the spread of available data. tackle has all these problems. We decided beforehand to remove the three main tubes. The main purpose is to display the input data at a fixed sample rate, in our case with 50 samples per second faster. The second blockchain, called Orientation-Independent Transformation (OIT), is used to represent data from multiple directions in a new space that does not depend on modern space and adapts to the weight and direction of travel.

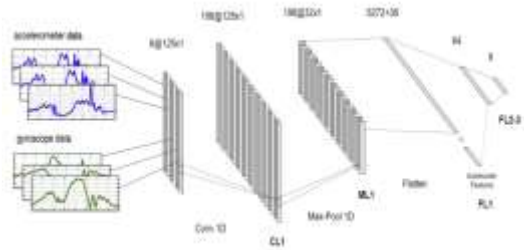

FIG 1.1: Structure of proposed system

This way users can place their smartphone anywhere, minimize space usage issues and rotate the smartphone, as we will see in Part VI. Our recent blog post describes a data collection operation that focuses on-axis symbols that are only used to engage to a certain degree. The reasons for this choice are explained. As previously reported [2], serial synchronization simplifies data entry and CNN simplifies data entry. Conversely, data manipulation should be avoided as it is useless in this case because it significantly distorts the timing of the sequence and removes large amounts of information critical for some operations. After improving distribution, we introduced a new teaching style. CNN includes the additional functionality of an automatic coding codec. As stated by Luga [2], CNN teaches that filters can be applied to a small area of data so that it can be in the form of information and environmental changes. due to the slight number of connections and a high amount of similarities, CNN's processing speed and execution time are considerably slower than other depth algorithms. This means that these examples are suitable for real-time HAR applications even in confined environments such as smartphones. One of the drawbacks of NCCs is that they delay the collection of symptoms. In [2] the author solves this tricky by adding some basic statistics to CNN. In addition to what we did in this last job using a number of handcrafted symbols, we decided to choose an auto-generated symbol. They have been created by an automatic encoder which offers a very sophisticated representation of the information. We can train the autoencoder separately and then use a trained format other than CNN.

Autoencoder

In this segment, we define the automatic encoder model based on [16], [17] and [18]. Autoencoder is a type of neural simulator used to train command data effectively without supervision. The job of the automatic encoder is to train the representation (coding) of certain information by instructing the network to ignore the noise signal. Given $X = [a_v, a_h, a_l, g_v, g_h, g_l]^T$ as a 6x125 matrix and x as the column vector 750x1 obtained by flattening X we can

define the autoencoder in two blocks. The first block is the encoder which is a function of the input:

$$f_\theta(x) = \sigma\left(W_2 f'_\theta(x) + b_2\right) \quad (1)$$

$$f'_\theta(x) = ReLU\left(W_2 x + b_1\right) \quad (2)$$

where θ_i is a vector of W_i and b_i which are weight matrix and layer polarization vector, respectively As a result of this block $y = f_\theta(x)$ can be created, The second block is the decoder, which starts with the code received from the encoder and reconstructs the input in the opposite direction

$$g_\theta(y) = W_4 g'_\theta(y) + b_4 \quad (3)$$

$$g'_\theta(y) = ReLU\left(W_3 y + b_3\right) \quad (4)$$

Thus, the reconstructed input is $z = g_\theta(y)$. What we want to minimize is the distance between x and z w.r.t. a distance measure with a harm function like MSE.

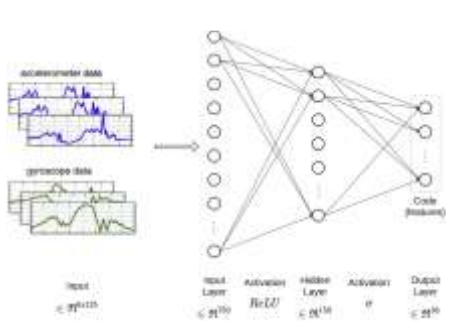

FIG 2.1: (Auto)-encoder structure

The main reason for using automatic encoder code for this app is to clear the payload as input for some operations. This is done using an automatic encoder code, i.e. execution of the block encoder:

$$\Phi_\varepsilon = y. \quad (5)$$

The autoencoder follows the structure reported in Fig. 2 and is based on a Neural Network architecture.

3.ALGORITHM

Convolutional Neural Network

Step 1: Load accelerometer data from the Heterogeneity data set.

Step 2: Convert and reformat accelerometer data into a time-sliced representation.

Step 3: Visualize the accelerometer data.

Step 4: Reshape the multi-dimensional tabular data so that it is accepted by Kera' s.

Step 5: Split up the data set into training, validation, and test set.

Step 6: Define a convolutional neural network model in Kera' s.

Step 7: Train the convolutional neural network for human activity recognition data.

Step 8: Validate the performance of the trained CNN against the test data using learning curve and confusion matrix.

Step 9: Export the trained Kera' s CNN.

Step 10: Create a playground in XCode and import the already trained Kera' s model.

4.RESULTS

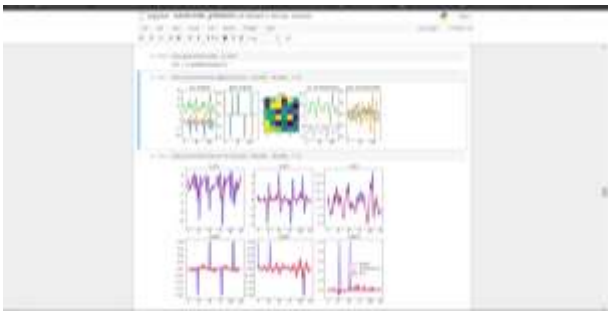

FIG 4.1: The autoencoder model

The above image represents the autoencoder model testing values in graph format.

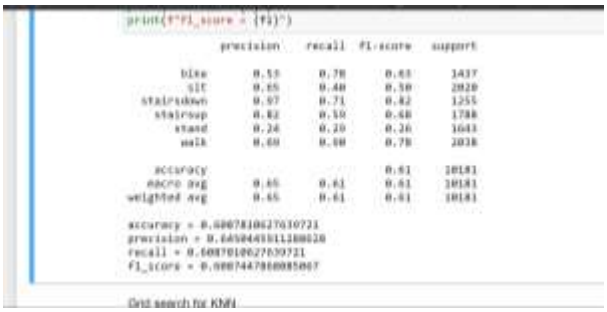

FIG 4.2: The autoencoder model and KNN classifiers with their evaluation

The above image represents the evaluation values of autoencoder and KNN Classifiers.

FIG 4.3: Confusion matrix

The above image represents the prediction accuracy in the format of confusion matrix.

FIG 4.4: Plot for NN based autoencoder;

The above image represents the NN based autoencoder plot image.

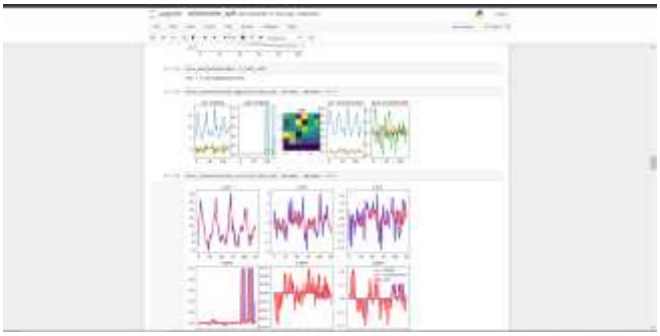

FIG 4.5: A test on splitted data (accelerometer/gyroscope);

The above image represents a splitted accelerometer and gyroscopic data in graph format.

FIG 4.6: The dataset pre-processing functions

Above image shows a pre-processing function for given dataset.

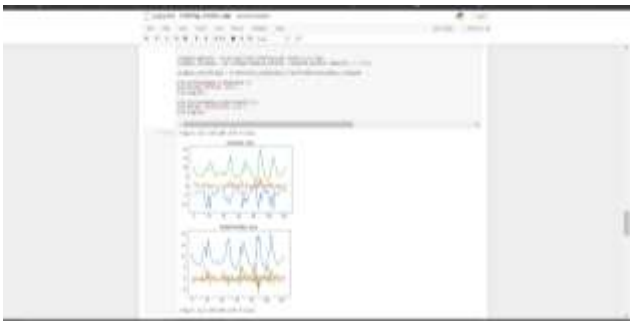

FIG 4.7: The CNN model and it's evaluation

Above image shows the evaluation of CNN model accuracy in the plot image.

5.COMPARATIVE STUDY

With this newly collected data set, we wanted to test the performance of our model in real use cases where smartphones can be placed in various positions and orientations. In this case, we train the Full HD model as the training set and then use OD as the test set. It is important to remember that in this case, we used the ILO and therefore reduced the HD exercise in sitting and standing to the inactive class category.

Autoencoder. An interesting finding is that the ILO is an essential operation in multiple orientations. For example, without OIT, we found that a trained and tested autoencoder with the same data went from 0.75 to 10.42 MSE: ten times worse. Apart from that, the details

Positions	Accuracy	Precision	Recall	F1-score
Pouch	80.1	86.4	80.7	79.0
Hand + Pocket	79.5	83.7	79.5	79.6
All	80.0	83.8	79.1	78.2

TABLE 1: KNN comparison between smartphone locations on OD.

Positions	Accuracy	Precision	Recall	F1-score
Pouch	74.4	84.7	74.4	74.8
Hand + Pocket	67.8	78.0	67.8	70.0
All	69.9	81.3	69.9	72.4

TABLE 2: Evaluation of the FFNN on the OD between smartphone positions.

CNN Network. By combining the automatic encoder function of CNN, we get better results with the simple KNN and FFNN classifiers above. For the new handheld + pocket job, we saw a nearly 25% drop in performance, but given the invisible complexity of this new context, we are quite satisfied with the results. Like Tab. 8 proves that the ultimate Smart preprocessing training strategy works well even with new task configurations similar to those used in real life.

We tried disabling OIT on CNN models, and the returns were reduced by almost 55% across all metrics. This shows that ILO is a very useful preprocessing technique for autoencoding and CNN modelling.

Positions	Accuracy	Precision	Recall	F1-score
Pouch	95.3	95.0	93.8	91.9
Hand + Pocket	80.5	89.5	78.0	80.2
All	98.6	97.9	96.1	96.1

TABLE 3: Comparisons of CNN classifications through smartphone positions on OD.

6.CONCLUSION

We offer a HAR execution solution that can be simulated in a mobile fitness application software, for example. It is very difficult to respond to this type of lag with a real mobile application and to set and customize it. With well-prepared pipelines, closely spaced lines, independent transformations, and location data, HAR losses due to new problems can be overcome. The literature suggests different ways of implementing a HAR: one of the advantages of using CNN is automatic property selection and the final ranking. By adding more advanced features to CNN, for example, B. By implementing an automatic encoder, we were able to extend the results before the original design, which used different data.

The following work can help demonstrate the effectiveness of this model by using more complex data from different types of people, such as weight, height, gender, etc., as well as the dissimilar types of services that destiny provides. It also helps create distinct shoes and clothes for the wearer. We also want to extend our visual activity to where the user is, e.g. cars, buses, trains, planes, etc. E. You may make other contributions because you don't have time. Confirm the use of open classification methods that require the system to reject detection/observation activities.

REFERENCES

[1] H. Blunck, N. O. Bouvin, T. Franke, K. Grønbæk, M. B. Kjaergaard, P. Lukowicz, and M. W¨ustenberg, “On heterogeneity in mobile

sensing applications aiming at representative data collection," in Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, pp. 1087–1098, 2013.

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: overview, challenges and opportunities," arXiv preprint arXiv:2001.07416, 2020.

[3] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," Applied Soft Computing, vol. 62, pp. 915–922, 2018.

[4] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," Pattern Recognition, vol. 74, pp. 25–37, 2018.

[5] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in Proceedings of the 13th ACM conference on embedded networked sensor systems, pp. 127–140, 2015.

[6] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," in 2011 International Conference on Body Sensor Networks, pp. 41–46, 2011.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting." in Journal of Machine Learning Research, 2014.

[8] K. Frank, M. J. Vera-Nadales, P. Robertson, and M. Angermann, "Reliable real-time recognition of motion related human activities using mems inertial sensors," in Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010), pp. 2919–2932, 2010.

[9] X. Gao, H. Luo, Q. Wang, F. Zhao, L. Ye, and Y. Zhang, "A human activity recognition algorithm based on stacking denoising autoencoder and lightgbm," Sensors, vol. 19, no. 4, p. 947, 2019.

[10] "Human activity recognition using smartphones data set." https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+ Using+Smartphones.

[11] "Wisdm's activity recognition using smartphones data set." https://www.cis.fordham.edu/wisdm/dataset.php.

[12] Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel, "Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–12, 2018.

[13] "The extrasensory dataset: A dataset for behavioral context recognition in-the-wild from mobile sensors." http://extrasensory.ucsd.edu/.

[14] "Heterogeneity activity recognition data set." https://archive.ics.uci.edu/ml/datasets/Heterogeneity+Activity+Recognition.

[15] F. Gu, K. Khoshelham, S. Valaee, J. Shang, and R. Zhang, "Locomotion activity recognition using stacked denoising autoencoders," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 2085–2093, 2018.

[16] C. R. Rao, "The use and interpretation of principal component analysis in applied research," Sankhy¯a: The Indian Journal of Statistics, Series A, pp. 329–358, 1964.

[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," Journal of machine learning research, vol. 11, no. 12, 2010.

[18] S. Duffner, S. Berlemont, G. Lefebvre, and C. Garcia,"3d gesture classification with convolutional neuralnetworks," in 2014 IEEE International Conferenceon Acoustics, Speech and Signal Processing (ICASSP) IEEE, 2014, pp. 5432–5436.

[19] Wang, Z.; Wu, D.; Gravina, R.; Fortino, G.; Jiang, Y.; Tang, K. Kernel fusion based extreme learning machine for cross-location activity recognition. Inf. Fusion **2017**, 37, 1–9.

[20] Ding, R.; Li, X.; Nie, L.; Li, J.; Si, X.; Chu, D.; Liu, G.; Zhan, D. Empirical Study and Improvement on Deep Transfer Learning for Human Activity Recognition. Sensors **2019**, 19, 57.

[21] Gupta, P.; Dallas, T. Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer. IEEE Trans. Biomed. Eng. **2014**, 61, 1780–1786.

[22] H. Abdi and L. J. Williams, "Goals of pca." in Principal Component Analysis, vol. 2. John Wiley & Sons, Inc., 2010.

[23] Laput, G.; Zhang, Y.; Harrison, C. Synthetic Sensors: Towards General-Purpose Sensing. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17), Denver, CO, USA, 6–11 May 2017; ACM: New York, NY, USA, 2017; pp. 3986–3999.

[24] Ordóñez, F.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for

Multimodal Wearable Activity Recognition. Sensors **2016**, 16, 115.

[25] Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. Pattern Recognit. Lett. **2019**, 119, 3–11.

[26] Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzek, M. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. Sensors **2018**, 18, 679.

[27] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones.," in Esann, vol. 3, p. 3, 2013.

[28] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am I facing: Inferring horizontal device orientation from an accelerometer signal," in 2009 International Symposium on Wearable Computers, pp. 149–150, 2009.

[29] Murad, A.; Pyun, J.Y. Deep Recurrent Neural Networks for Human Activity Recognition. Sensors **2017**, 17, 2556.

[30] Peng, L.; Chen, L.; Wu, M.; Chen, G. Complex Activity Recognition using Acceleration, Vital Sign, andLocation Data. IEEE Trans. Mob. Comput. **2018**.