Domain Human Recognition Techniques using Deep Learning

Dr. Seshaiah Merikapudi[1], Dr. Murthy SVN[2], Manjunatha.S[3], R.V.Gandhi[4]
Associate Professor, Department of CSE, SJC Institute of Technology, Chickballapur, India[1, 2, 3]
Assistant Professor, Department of CSE(DS)[4]
Keshav Memorial Institute of Technology, Hyderabad, Telangana, India[4]

***Abstract*—As a key research subject in the fields of health and human-machine interaction, human activity recognition (HAR) has emerged as a major research focus over the past few decades. Many artificial intelligence-based models are being created for activity recognition. However, these algorithms are failing to extract spatial and temporal properties, resulting in poor performance on real-world long-term HAR. A drawback in the literature is that there are only a small number of publicly available datasets for physical activity recognition that contain a small number of activities, owing to the scarcity of publicly available datasets. In this paper, a hybrid model for activity recognition that incorporates both convolutional neural networks (CNN) are developed. The CNN network is used for extracting spatial characteristics, while the LSTM network is used for learning time-related information. Using a variety of traditional and deep machine learning models, an extensive ablation investigation is carried out in order to find the best possible HAR solution. The CNN approach can achieve a precision of 90.89%, indicating that the model is suitable for HAR applications.**

Keywords—Human recognition; deep learning; hybrid model; CNN; HAR

I. Introduction

Human activity recognition (HAR) is a well-established research topic that requires the correct identification of a wide range of activities that are collected in a variety of ways. Sensor-based HAR makes use of inertial sensors such as accelerometers and gyroscopes to measure the acceleration and rotational velocity of a body. There are numerous advantages to employing sensors to capture a person movement rather than cameras and microphones, including the fact that they are less sensitive to noise, less invasive for the user, and less expensive. Sensors are also less expensive than cameras and microphones [1]-[5]. Furthermore, as a result of the growing use of sensors embedded in cellphones, these devices have become virtually ubiquitous in our lives.

Aspects of sensor-based HAR that are particularly challenging are the encoding of information and the representation of time. Previous classification systems depended on features extracted from kinetic signals that were first constructed and then implemented into the system. Please keep in mind that these characteristics are largely picked on the basis of heuristics, which are determined by the nature of the work at hand. If you have a deep grasp of the application area or extensive human experience, you may find that when you extract the characteristics from the data collection, you only get a shallow set of characteristics. As previously stated, standard HAR techniques do not scale well to complex motion patterns and do not perform well on dynamic data, which is defined as data gathered from streams that remain forever outside of the lab [6]-[9].

Automated and deep approaches to human-computer interaction are becoming increasingly prevalent in the field of human-computer interaction. Most deep learning research in biometrics has been focused on face and speaker recognition [12]. The selection of significant characteristics from the data is delegated to the learning model through the use of data-driven signal classification methods, which are used to train the learning model on the data. CNNs are particularly useful when it comes to detecting spatial and temporal correlations between signals [10]. Efficient features are first extracted from raw data. The features include mean, median, autoregressive coefficients, etc. [11].

This paper presents the construction of an activity recognition model that incorporates convolutional neural networks (CNNs). The CNN network is used to extract spatial features, whereas the LSTM network is used to learn about time-related information. It is necessary to do a thorough ablation analysis utilizing a variety of classical and deep machine learning models in order to determine the most effective HAR solution possible. The CNN technique can be employed for HAR applications because of its high precision of 90.89%.

II. Related Work

When it comes to common supervised machine learning techniques, the generic Activity Recognition Chain [13] includes steps such as preprocessing, extraction of features, and classification. When it comes to deep learning (DL), CNNs are an example of a technique that does not require the extraction of features from raw data before classification [14]. CNN feature extraction is accomplished through the convolution of the input signal with a kernel [15]. The outcome of the convolution technique is a feature map that contains information about the data.

Both advantages and drawbacks arise from the ability of CNNs to automatically learn properties. It streamlines the ARC [14] by automating jobs that would otherwise require domain-specific expertise, like the identification of a suitable feature collection. In contrast to the feature selection approach, which starts with the largest number of features feasible and narrows it down to only those that provide the best discrimination across target classes, CNNs do not require any of these phases. Instead, the use of a CNN incorporates the feature extraction phase into the classifier model, requiring an extended training period to generate adequate features and exposing the approach to problems such as those arising from cold starts. In order to mitigate this problem, it is common in computer vision to use pre-trained CNN models for feature extraction, such as those developed by Raja Raman et al. [16].

In the MLP classifier, there are numerous dense, fully connected layers that lead to an output layer with as many n-nodes as the number of target classes present in the input. A vector of HCF is used as an input to a regular MLP in order to feed it (a). In the CNN scenario (b), convolutional layers are employed to extract features from the data [17].

An alternative is to do a max-pooling operation after each of the convolutional layers, which will result in a further reduction in the feature map size due to down sampling of the data [17]. A 1D vector that has been flattened from the output of the previous convolutional layer is supplied into the MLP layer just as it was in the HCF example. While IMU signals frequently contain a temporal component, 2D convolution is more commonly used in the processing of picture audio [18] and video audio [19].

Rectified Linear Unit (ReLU) activation functions are one of the most frequently used activation functions for convolutional layers, while Softmax activation functions are commonly applied in multi-class classification settings [15]. It is possible to employ alternative activation functions in the case of multi-label classification, such as the sigmoid function [20].

III. Proposed Method

A two-pronged approach will be used to experiment with CNN in this section. In the first phase of this work, CNN automatically retrieved features with a variety of hyper parameters and topologies, which are then analyzed and the results were presented. A real-world dataset was utilized for the second aim, which investigated the viability of employing a pre-trained CNN feature extractor for HAR. It is useful to take advantage of a CNN ability to automate feature extraction while avoiding the cold-start difficulty.

Two steps in a case study were recommended, which are depicted in Fig. 1.

In the first stage, a CNN feature extractor is trained to extract features from images [Fig. 1(a)]. This step involves experimenting with different topologies and hyper parameter combinations. The best-performing HAR models have been discovered as a result of this research. It is only used as a feature extractor in the second stage, in order to convert raw data into a suitable input vector for a second classifier model in the third phase. The flattening layer generates feature vectors as a result of a succession of convolutional processes, and the weights of the CNN networks are fixed at the beginning of the first phase.

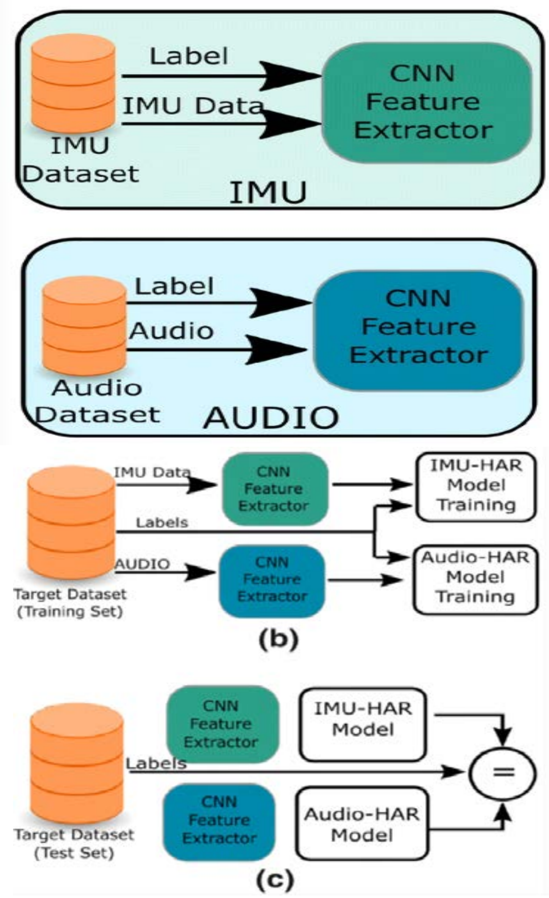

Fig. 1. Two-pronged Approach will be used to Experiment with CNN.

Following an aim similar to transfer learning, the feature vector formed by taking the output of the flattening layer can be used to represent raw data in a different context by using the feature vector created by taking the output of the flattening layer. As seen in Fig. 1(b), the characteristics generated by the pre-trained classifier are utilized to train the second model, which is subsequently used to train the first model. Finally, as illustrated in Fig. 1(c), the second model is put to the test.

In several preliminary experiments, HCF and CNN were compared. However, the breadth of the current case study was restricted due to the nature of the data. For example, CNN feature extractor for HAR is used in this work to explain how to use the tool, and some additional data and analysis are included that was not included in the previous study to demonstrate how to utilize the tool.

As a result of the prior investigation, several key data requirements for the case study were identified. For the first phase, it was found that data collected in controlled environments was the most appropriate choice. The likelihood of noisy labels being introduced into these datasets is low due to the fact that they are developed in a controlled laboratory environment. Consider that the comparison with HCFs may be influenced by issues such as label noise, which could result in an incorrect rating.

A. Network Architecture

Because each sensor samples six different signal components, the type of input examples that the network receives is determined by the sensor design. These six different signal components are then organized into a single channel image matrix of size 6×204×N, which is the largest size available. Consequently, the network input takes the shape of 6×204×N, where N is the number of channels and is equivalent to the number of sensors used for sampling in the network. Model-driven input adaptation is the term used to describe this technique, which is capable of recognizing both spatial and temporal patterns within the signal components.

The convolutional model is illustrated in its entirety in Fig. 3. Following the input layer, there are three convolutional layers and three max-pooling layers. Following this procedure, each input channel receives a collection of multiple feature maps, each with kernels of sizes 3×5, 2,4×2, and 2×2 according to the size of the input channel. It is necessary to pad the input of each convolutional layer correctly in order to ensure that there is no resolution loss caused by the convolutional process. A batch normalization procedure is employed during the creation of each convolutional layer. In the three max-pooling layers, kernels with sizes of 3 3, 2×2, and 3×2 are employed. Following that there are three dense layers of 500, 250, and 125 units apiece, which together form a network that is entirely connected. During the training phase, neurons have a 0.5% chance of being dropped from the thick layers.

It is true that the cross-entropy function is used to measure loss; however, it is also used as an activation function for all of the network nodes as well. The Adam optimizer is a stochastic approach to optimization that uses a random number generator. Units in the output layer correspond to the number of actions performed by each group, and there are m units in total. The softmax method will return the class of the input windows that is the most likely to be encountered.

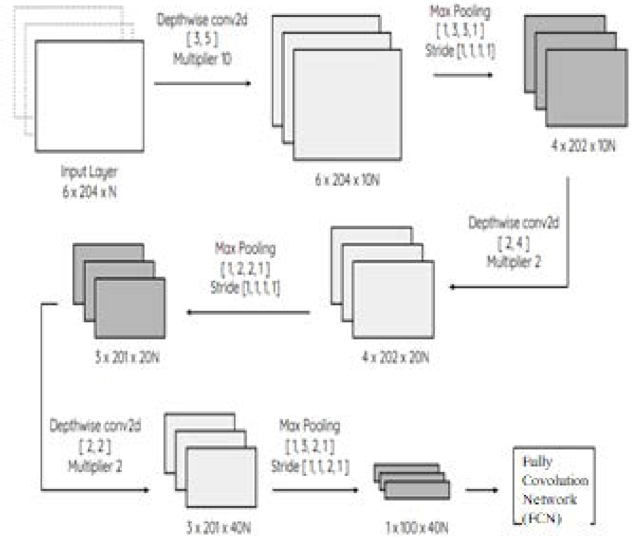

Fig. 2. CNN Architecture.

The Fig. 2 represents the CNN Architecture. The input results in fully convolutional network (FCN).We have chosen a set of hyper parameters that are consistent across all activity groups and sensor configurations. These parameters were chosen based on best practices in the literature and empirical evidence. It has been discovered that a batch size of 1024 can significantly speed up the learning process when compared to smaller batches, while not being computationally prohibitively complex to maintain.

Depending on the behavior of specific combinations, there are between 150 and 300 training epochs available. The starting rate of learning is set at 0.005% each minute. Rather than attempting to construct the most efficient network, our goal is to assess the classification potential of various sensor technologies.

We make our architectural decisions as a result of a very normal network configuration, which is based on modest kernels, standard regularisation methods, and a small number of hyperparameters. If no regularisation process is employed, three convolutional layers will result in overfitting if no regularisation procedure is used. The inclusion of extra convolutional or dense layers has little effect on the performance of the network, but the introduction of dropout helps to stabilise learning and improve stability.

IV. Results and Discussion

The Tensor Flow 1.7 framework is utilized in the construction of the network. Following the recording of the activity, the signal is decomposed into 204 points, each of which corresponds to roughly two seconds of movement, with a stride of five points between each point. Reduced window sizes are associated with enhanced classification performance, and in the context of CNNs, the limited structure of the network input data makes the training process easier. The shape of the generated matrix is determined by the number of sensors employed to sample the window: 6N×204 in this case. This has resulted in a significant increase in the quantity of training and testing samples available. It is uneven because the activities have varying execution periods and because different subjects may complete the same activity at various rates, which makes the dataset unbalanced.

For the purpose of evaluating the performance of our classification system, we employ a typical 5-fold cross-validation approach. To separate the accessible datasets, we employ topics rather than windows as a division method. As a result, this prevents overfitting and enhances the generalization of model outputs as a result. In order to produce each fold, an 80/20 split is achieved by separating four people from a group of 19, as mentioned previously.

Through this case study, which is shown in Fig. 3 to Fig. 7 were able to investigate the impact of various hyper parameters and CNN settings on the feature learning capabilities of our model. Regarding feature learning in HAR, this experiment provided an excellent overview of the main elements that determine a CNN ability to learn new features and how these aspects interact with one another. While the results from the first phase of the case study demonstrated that CNNs can perform at least as well as the best HCFs, the results from the second step demonstrated that CNNs can perform at least as well as the best HCFs. This is why CNNs must be trained before being used, and as a result, they are susceptible to the cold-start problem.

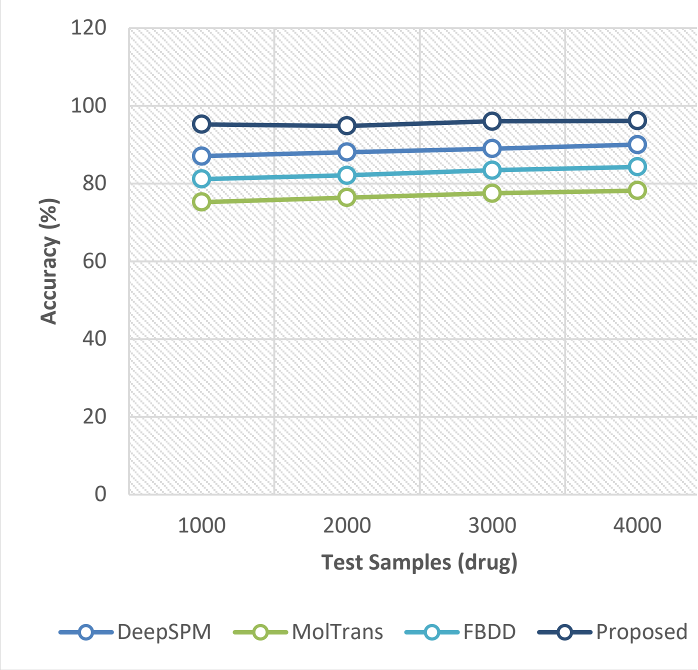

Fig. 3. Accuracy.

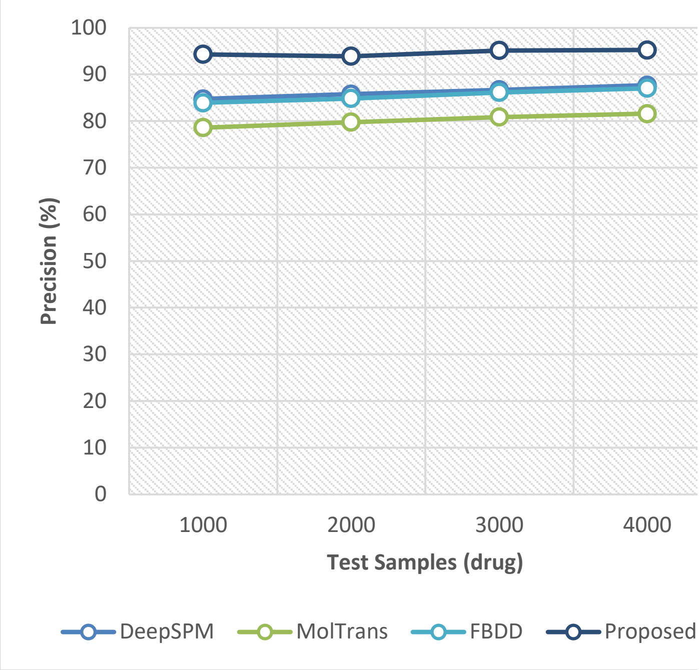

Fig. 4. Precision.

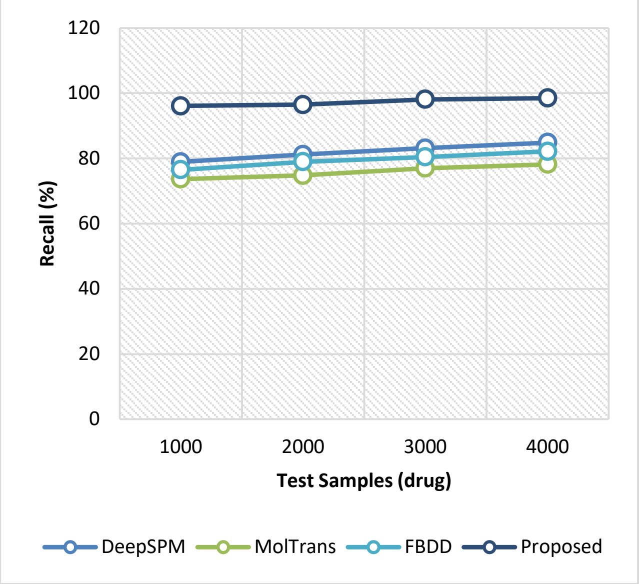

Fig. 5. Recall.

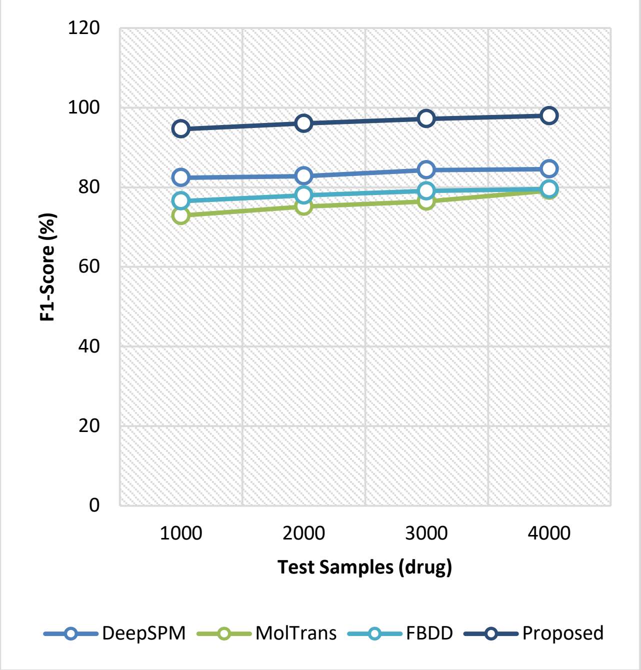

Fig. 6. F1-Measure.

To evaluate CNN feature learning methods for dataset test cases, a pre-trained CNN feature extractor was employed on a real-world dataset to extract features from it. Real-world datasets enable the evaluation of CNN automatic features in a more realistic environment than was previously possible. The problem of dealing with uncontrolled environments was brought to light through realism-based testing.

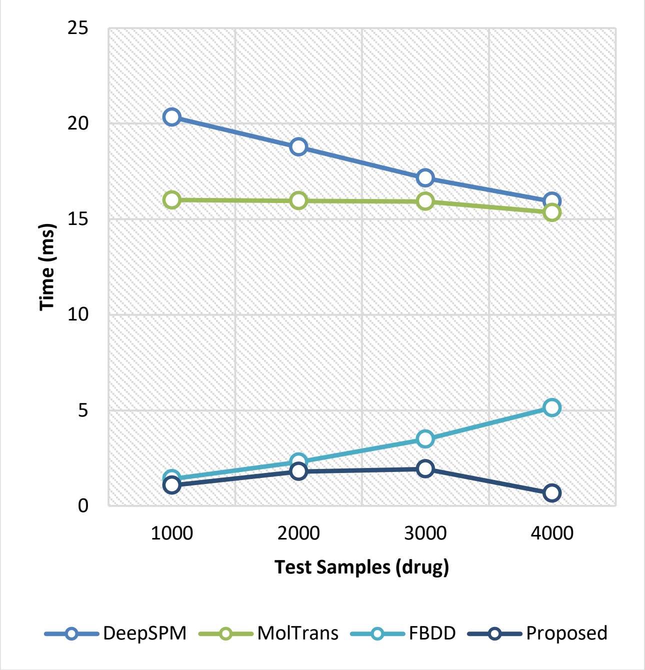

Fig. 7. Computational Time.

Although a larger number of training epochs were required, it was demonstrated that SGD enabled the model to be trained with greater accuracy on the test set, leading to an F-Score of 94% in the more challenging target activities, which included transitions as a NULL class. We were successful in identifying a suitable network architecture that matched our feature learning criteria while also keeping the model complexity under control. The architecture was evaluated using data from a large real-world dataset that was made available to the public.

V. Conclusion

This paper presents the construction of an activity recognition model that incorporates both convolutional neural networks (CNNs) and convolutional neural networks (CNNs). This work was primarily intended to demonstrate the usage of a CNN pre-trained feature extractor rather than to provide a comprehensive analysis of the hyper parameters and settings used in the training process. The optimization of models, on the other hand, is subject to several restrictions. When analyzing designs with varied numbers of convolutional layers and convolution kernel sizes, for example, only the ReLU activation function was used to achieve the best results.

Other modes of activation may be investigated in future investigations. In a similar vein, the default Keera’s learning rate of 0.001 was employed. The CNN network is used to extract spatial features, whereas the LSTM network is used to learn about time-related information. It is necessary to do a thorough ablation analysis utilising a variety of classical and deep machine learning models in order to determine the most effective HAR solution possible. The CNN technique can be employed for HAR applications because of its high precision of 90.89%.

References

[1] Ramasamy Ramamurthy, S., & Roy, N. (2018). Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1254.

[2] Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., & Liu, Y. (2021). Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, *54*(4), 1-40.

[3] Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, *105*, 233-261.

[4] Zhang, J., Yin, Z., Chen, P., & Nichele, S. (2020). Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. *Information Fusion*, *59*, 103-126.

[5] Antar, A. D., Ahmed, M., & Ahad, M. A. R. (2019, May). Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 134-139). IEEE.

[6] Rashid, K. M., & Louis, J. (2019). Times-series data augmentation and deep learning for construction equipment activity recognition. *Advanced Engineering Informatics*, *42*, 100944.

[7] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, *51*(5), 1-36.

[8] Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., & Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 206-219.

[9] Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. *ACM Computing Surveys (CSUR)*, *52*(4), 1-35.

[10] Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. (2019). Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, *7*, 7327-117345.

[11] Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, *81*, 307-313.

[12] Sundararajan, K., & Woodard, D. L. (2018). Deep learning for biometrics: A survey. *ACM Computing Surveys (CSUR)*, *51*(3), 1-34.

[13] Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, *46*(3), 1-33.

[14] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[15] Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115.

[16] Rajaraman, S., Antani, S. K., Poostchi, M., Silamut, K., Hossain, M. A., Maude, R. J., ... & Thoma, G. R. (2018). Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ, 6, e4568.

[17] Baldominos, A., Cervantes, A., Saez, Y., & Isasi, P. (2019). A comparison of machine learning and deep learning techniques for activity recognition using mobile devices. *Sensors*, *19*(3), 521.

[18] Moya Rueda, F., Grzeszick, R., Fink, G. A., Feldhorst, S., & Ten Hompel, M. (2018, June). Convolutional neural networks for human activity recognition using body-worn sensors. In *Informatics* (Vol. 5, No. 2, p. 26). Multidisciplinary Digital Publishing Institute.

[19] Saeed, A., Ozcelebi, T., Trajanovski, S., & Lukkien, J. (2018). Learning behavioral context recognition with multi-stream temporal convolutional networks. *arXiv preprint arXiv:1808.08766*.

[20] Huang, S. J., Gao, W., & Zhou, Z. H. (2018). Fast multi-instance multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, *41*(11), 2614-2627.