A CNN-LSTM Approach to Human Activity Recognition

Ronald Mutegeki
School of Computer Science and Engineering
Kyungpook National University
Daegu, Korea
rmutegeki@knu.ac.kr

Dong Seog Han *
School of Electronics Engineering
Kyungpook National University
Daegu, Korea
dshan@knu.ac.kr*

***Abstract* — To understand human behavior and intrinsically anticipate human intentions, research into human activity recognition HAR) using sensors in wearable and handheld devices has intensified. The ability for a system to use as few resources as possible to recognize a user's activity from raw data is what many researchers are striving for. In this paper, we propose a holistic deep learning-based activity recognition architecture, a convolutional neural network-long short-term memory network (CNN-LSTM). This CNN-LSTM approach not only improves the predictive accuracy of human activities from raw data but also reduces the complexity of the model while eliminating the need for advanced feature engineering. The CNN-LSTM network is both spatially and temporally deep. Our proposed model achieves a 99% accuracy on the iSPL dataset, an internal dataset, and a 92% accuracy on the UCI HAR public dataset. We also compared its performance against other approaches. It competes favorably against other deep neural network (DNN) architectures that have been proposed in the past and against machine learning models that rely on manually engineered feature datasets.**

Keywords—Human activity recognition (HAR); Convolutional neural network (CNN); Long short-term memory network (LSTM); CNN-LSTM; deep learning; UCI HAR dataset;

I. INTRODUCTION

Human activity recognition (HAR) based on inertial measurement unit (IMU) has become the de facto method for continuously monitoring not only what human beings are up to but also in monitoring the activities of devices, machine parts, pets, and others. This has made HAR based on IMU sensors a hot area for research. Not to mention that these maintain high levels of privacy and comfort for the user. Much as some approaches to correctly classifying the activities of a user with IMU sensor data have been proposed [5, 6, 8], many of them make it out to be a very difficult task. Many require a lot of resources, domain expertise among other barriers. The emergence of deep learning into the field of HAR has made the task of activity recognition very trivial. Hammerla *et al.* [1] claim that deep learning represents the biggest trend in machine learning over the past decade. Machine (ML) and deep (DL) learning models are readily made available by frameworks like TensorFlow, PyTorch, Scikit, and others not to mention the Keras API [2] that has made it easy to build and experiment with models.

With the current advances in other fields using deep learning, a field that hasn't received as much attention is HAR. In a typical HAR scenario, a user with a device (could be a standalone sensor, smartwatch, smartphone, etc.) that is equipped with gyroscope and accelerometer sensors, continuously sends sensor data to a listening server that enables continuous activity monitoring of the user. Variations to this architecture exist, especially with modern smart devices having the capabilities to perform activity recognition and monitoring on their own. These have better processing units, larger memories, and better sensors.

With deep learning, it has become a lot easier to train a model to recognize certain activities from raw sensor data in a fast and efficient way. Some of the existing machine learning algorithms that have become near obsolete include the support vector machine (SVM) used in [6], the histogram of gradients (HOG) feature extraction with a k-nearest neighbour classifier [5] that have been proposed in the past. These approaches managed to reach impressive recognition accuracies. However, a lot of work was needed to prepare the data, domain knowledge, pre-processing, among others.

Enter deep learning, where given a raw sensor signal, a deep learning model can extract features and make predictions in an efficient manner. Some approaches that have been proposed include convolutional neural networks (CNN) [9] which are spatially deep, long short-term memory (LSTM) networks which are temporally deep, deep feed-forward neural networks and their variations. Each of these approaches has its strengths and weaknesses. Furthermore, they have been designed with specific application goals in mind. However, we noticed that we could leverage the strength of one network to improve on the robustness of the other.

In this paper, we present a CNN-LSTM classifier for human activity recognition. Whereas both CNN and LSTM networks have been extensively researched in the past, they've been studied in isolation. Our paper seeks to leverage the strengths of combining the two networks especially in as far as human activity recognition is concerned. In Section II, we take a look at the background of our study and review some previous work. Section III discusses our method and its implementation. We take a look at the performance of the CNN-LSTM model in contrast with other models on both the intelligent Signal Processing Lab (iSPL) dataset and the UCI HAR dataset in Section IV. We finally conclude this research work in Section V.

II. BACKGROUND AND PREVIOUS WORK

There exists a lot of research into human activity recognition using a multitude of approaches as illustrated by Anguita *et al.* [6], Eyobu and Han [8] and many others. In this section, we set a foundation for deep learning based HAR while exploring some of the previous work that relates to our proposed approach and how the works differ from our approach.

Human Activity Recognition and Datasets

The current HAR problem aims at using sensors; accelerometer, gyroscope, magnetometer, and others; that are built into IMU devices, and smartphones to recognize the activity being performed by the user of the device [3]. The multimodal signals from these sensors are collected over time, which makes them time dependent. We, therefore, refer to the HAR problem as a time series classification (TSC) problem.

Ismail *et al.* [4] defined a time series $X = [x_1, x_2, \dots, x_T]$, as an ordered set of real values and its length is the number of

real values T. He goes further to define a dataset $D = \{(X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)\}$ as a collection of pairs (X_i, y_i) where X_i is a time series with y_i as its corresponding label or class. A model is then built to classify the dataset D so as to map the space of possible inputs X_i to a probability distribution over the class variable values Y_i which forms the foundation of any machine learning model for a TSC problem. Since HAR is a TSC problem, several deep learning and non-deep learning approaches have been suggested. To note a few; the multi-class SVM suggested by Anguita *et al.* [6], LSTM [8], CNN [4, 9], among others.

In this paper, we use two datasets to evaluate the performance and applicability of our proposed approach. The first is a relatively small dataset we built in our lab that features 3 activities of Standing, Sitting, and Walking referred to as the iSPL dataset. This dataset was specific to the needs of the experiments we were conducting that involved a multi-user indoor localization system with human activity recognition. The second dataset we used is the University of California Irvine human activity recognition (UCI HAR) using smartphones dataset version 1 from [6]. This dataset has been an outstanding benchmark for smartphone and wearable sensor-based activity recognition tasks and has been used in several works [3, 5, 6, 8]. In our previous work [3], we used it to evaluate transfer learning where we split the dataset into 2 different tasks. Jain and Kanhangad [5] evaluated the performance of different feature engineering techniques using machine learning on this very dataset. Eyobu and Han [8] used this dataset to evaluate the performance of their proposed feature extraction and data augmentation techniques. All these works significantly differ from our proposed work since the goals differ significantly. More information about the datasets we used is provided in Section III.

Deep Learning for HAR

Machine learning (ML) has been at the core of research into human activity recognition for a very long time [6]. However, since AlexNet [7] won the ImageNet competition in 2012, deep learning has seen successful applications in a multitude of domains. Computer vision, natural language processing, speech recognition, etc. This success led several researchers to use various deep learning approaches in solving the HAR problem [6, 8, 9, 10]. In our research, we take a deep learning approach to HAR in a way that seeks to reduce on the efforts needed for feature engineering which requires domain knowledge.

In a typical ML approach, triaxial data from a user wearing a device equipped with the requisite sensors; accelerometer and gyroscope, is collected, windowed into samples, and statistics in the frequency domain, and time domain or both are obtained [6]. These statistics, as opposed to raw data, are used as the features from which we infer the activity to which a given sample belongs. This approach gives very impressive results but heavily relies on the feature extraction technique used, the distribution of the data, the sensor orientation, and fares quite badly in an environment outside that which it was trained. Additionally, the models trained with this kind of approach tend to fare badly when adapted for a different device. Instead of learning from sections of the signal itself, the model is trained on aggregates of the entire signal thus failing to adapt to when slight changes in the same signal occur. The temporal nature of the HAR problem too presents a huge challenge to the ML approach and thus calling for DL.

CNN and LSTM networks for HAR

Convolutional Neural Networks (CNN) are prevalent in image recognition tasks [7]. Hammerla *et al.* [1] suggested that CNNs were able to improve on the state-of-the-art performance in several areas and has also been used for HAR and Ubiquitous Computing. This approach to HAR is quite popular but has found much more success in video and image data as opposed to sensor data. CNNs are networks that are known to be spatially deep. In our research, we rely on this property to aid in extracting the spatial features of the signal.

Hammerla *et al.* [1] go ahead to suggest that approaches that can exploit the temporal dependencies in time-series data appear as the natural choice for modelling human movement captured with sensor data. Deep recurrent networks, most notably those that rely on Long Short-Term Memory cells (LSTMs) [12], have recently achieved impressive performance across a variety of scenarios. This temporal characteristic of the LSTM architecture and its long-term dependences make it a solid candidate to extract temporal features from our signal. Our work differs from [1] in a way that while Hammerla *et al.* [1] were comparing the performance of individual networks for HAR, our work seeks to leverage the combined power of both networks.

Eyobu and Han [8] employed an LSTM network to classify a dataset with frequency domain features extracted, as opposed to the raw signal which differs from our approach. Kim *et al.* [10] employed a CNN-LSTM hybrid model that consisted of 3 convolutional layers, 2 LSTM layers with 128 hidden units each and a softmax classifier for 2 classes. It is then validated on their own dataset and the performance in terms of error and classification accuracy compared with other machine and deep learning models; SVM, LSTM, etc. In our experiments, we chose to use another implementation of the CNN-LSTM ensemble classifier that is explained in Section III, whose architecture and objectives clearly differ from [10].

III. METHOD AND EXPERIMENT SETUP

In our experiments, we use the iSPL dataset and the UCI HAR dataset that is explained by Anguita *et al.* in [6] both mentioned in Section II above. The iSPL dataset is a 3-activity dataset that contains triaxial raw accelerometer and gyroscope signals collected from a WithRobot™ sensor strapped to the left-hand wrist of a subject. There were 4 subjects aged between 25 and 40 years. The dataset is comprised of a total of 1,590 samples that are split randomly into 1,272 and 318 train and test samples respectively. Each data sample is comprised of 128 sensor readings for each of the 9 signal types elaborated below.

The UCI HAR dataset is a 6-activity dataset that contains 3D (x, y, z) raw signals extracted from the accelerometer and gyroscope of a smartphone strapped to the waist of a subject [6]. The experiments were carried out with a group of 30 subjects within an age bracket of 19-48 years. Each person performed six activities of Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, and Laying. The dataset is comprised of 7,352 and 2,947 train and test samples respectively. As elaborated in [6], the sensor signals were pre-processed by applying noise filters and sampled in fixed-width sliding windows of 2.56 sec with a 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity.

Fig. 1 below shows a block diagram of our proposed CNN-LSTM approach we use when recognizing and classifying activities. Chollet [2] suggests that the Keras API enables one to go from idea to result with the least possible delay and this being key to doing good research. We built a Keras sequential model with a TensorFlow backend. The model takes in 9 signals; total acceleration (a_x, a_y, a_z), angular velocity from the gyroscope (g_x, g_y, g_z) and body or linear acceleration without the acceleration attributed to gravity (la_x, la_y, la_z).

For experiment purposes and to maintain a controlled experiment, we chose 1 1D convolution layer (tf. keras. layers. Conv1D) with 64 filters and kernel_size =3 which specifies the length of the 1D convolution window. This layer uses a ReLU activation function. We left the rest of the parameters as defaults. Next, we added a 1D maxpooling layer with pool_size set to 2 and a flatten layer whose job is to format the feature data from this section to be consumed by the LSTM layer in the next step. The nature of the convolution layer is in such a way that the data it works with is quite different from the data that is accepted by the LSTM layer. The temporal nature of the data that we are dealing with too calls for the use of a mechanism to handle this challenge. Keras provides a TimeDistributed wrapper that takes in a layer as an argument and applies convolutions to the signal while maintaining its temporal integrity for the LSTM layer(s)[11]. Since the input to a TimeDistributed layer should at least be 3D, we reshaped out input signal from 128 time steps with 9 signals (dimensions/channels) for each sample to 4 slices each of 32 time steps. (None, 128, 9) was transformed into (None, 4, 32, 9) to be consumed by a TimeDistributed 1D convolution layer. All layers before the LSTM layer(s) were all wrapped in this TimeDistributed wrapper.

The flatted feature maps from the previous layers are then used as input to an LSTM layer with 128 units and a ReLU activation. The LSTM layer(s) extracts the temporal dependencies of the signal. Signal data is sequential in nature and the best models that can handle such data belong to the recurrent neural network (RNN) category that the LSTM network belongs to. Several advantages that come with using the LSTM network over other deep neural networks have been elaborated in [1]. All the default configurations of the layer are left intact except for the number of units that we set to 128 [12]. This layer uses a *tanh* activation.

The output from the LSTM layer is then sent to a fully connected (FC) output layer with a Softmax activation which classifies the given input into the given number of classes. In our experiments, we had a 3-class classification for the iSPL dataset and a 6-class classification for the UCI HAR dataset.

In comparing the performance of this approach when compared to other existing architectures and approaches, we used variations in this design. In one variation, we added a 50 neuron Dense (Fully connected) layer between the LSTM layer and the output Softmax layer. We refer to this model as the "CNN_LSTM_Dense" since it connects the CNN_LSTM model to a Dense layer. The results of this model compared to others are presented in the next section and are fully discussed.

In the second variation, we removed the convolution layer and added a second LSTM layer. The first LSTM layer returns sequences using the return_sequences flag set to true. Both layers have been configured with 128 hidden units with a *tanh* activation function and a hard_sigmoid as the recurrent layer's activation. We refer to this network as just the "LSTM" network since it is comprised of only LSTM layers save for the output Softmax layer.

In the third and final variation, we added a fully connected dense layer to the network above just like we did for the CNN_LSTM. We refer to this network as the "LSTM_Dense" network. The FC layer has a ReLU activation. We configured it to have 50 hidden neurons just like in the CNN_LSTM_Dense network above.

A number of the hyper parameters that we chose were left constant across all models but as far as we know, these hyper parameters were optimal and had little impact on the performance of the respective models as a number of repeated experiments with different hyper parameters showed a very slight change in the performance of the given models. The learning rate, lr, was set to 0.0025, a mini-batch size of 64, and 30 training epochs were some of these hyper parameters.

IV. Experimental Results And Discussion

We ran several experiments to thoroughly evaluate the performance of the models explained above to confirm whether the CNN_LSTM model would perform as anticipated. We used the 2 datasets explained in Section II above to perform different experiments on the models and the results are illustrated in subsequent sections.

Fig. 2 below shows the accuracy attained from the various models we trained on the iSPL dataset which has 3 classes.

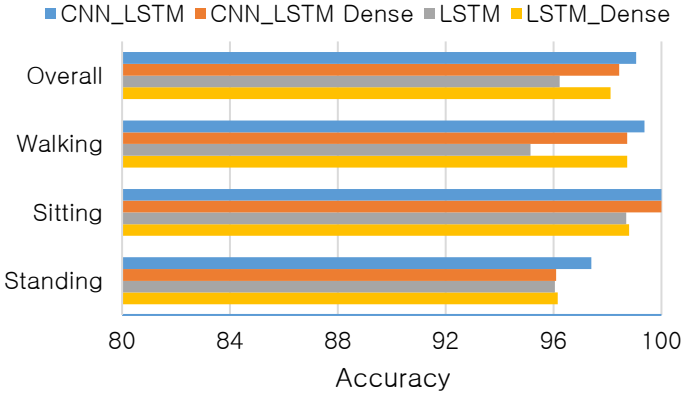

Fig. 2 Bar chart showing accuracy of the different models on the iSPL dataset

We see that the CNN_LSTM model outperforms all the other models with 99.06%. It is closely followed by the CNN_LSTM_Dense model with 98.43%. The best classified

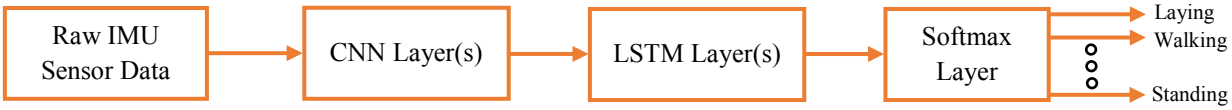

Fig. 1 Block diagram of the proposed CNN_LSTM architecture for human activity recognition

activity is Sitting with the 2 models achieving 100% accuracy. The model with the worst overall performance is the LSTM model with 96.23% followed by 98.11% for the LSTM_Dense model. This has been illustrated further in TABLE 1.

Fig. 2 below shows the accuracy attained from the various models we trained on the UCI HAR dataset with 6 classes.

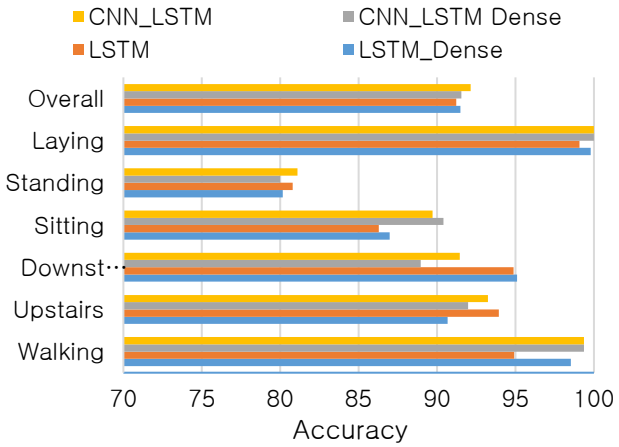

Fig. 3 Bar chart showing accuracy of the different models on the UCI dataset

Just like for the iSPL dataset, the CNN_LSTM model outperforms all the other models at 92.13%. It is followed by the CNN_LSTM_Dense model with 91.55%. The best classified activity is Laying down with 3 of the 4 models achieving over 99% accuracy. The model with the worst overall performance is LSTM with 91.23% followed by 91.48% for the LSTM_Dense model. This has also been illustrated further in TABLE 1. From this result, we can see a similar trend like that in the iSPL dataset which indicates that the CNN_LSTM mode performs better than its peers.

TABLE I. ACCURACY OF THE CNN-LSTM MODEL ON BOTH THE ISPL AND UCI DATASETS

Model	iSPL	UCI
CNN_LSTM	**99.06%**	**92.13%**
CNN_LSTM_Dense	98.43%	91.55%
LSTM	96.23%	91.28%
LSTM_Dense	98.11%	91.40%

Another metric we used to evaluate the performance of our proposed model vis-à-vis other models is the categorical Cross-Entropy (loss). In our experiments, we are trying to categorize a given sample as belonging to one of many classes (activities) which is a multi-class classification problem. The model evaluates how accurately a sample has been placed in its right class by associating a probability to each output neuron/class. We refer to this loss as the Softmax loss which is the Softmax activation in equation (1) plus a Cross-Entropy loss in equation (2) below: [13]

$$f(s)_i = \frac{e^{s_i}}{\sum_i^C e^{s_j}} \quad (1)$$

$$CE = \sum_i^C t_i \log(f(s)_i) \quad (2)$$

where t_i and s_i are the ground truth and the model's score for each class i in C. Gómez [13] goes ahead to give more information regarding this loss function and how it is derived.

Fig. 4 below illustrates the Softmax loss for the different models we were evaluating. Just like it was in evaluating the accuracy, the CNN_LSTM model has the lowest Softmax loss value of 3.92% and is closely followed by the CNN_LSTM_Dense model at 5.77%. The worst loss of 13.84% for the LSTM model is followed by the LSTM_Dense model's 10.81% loss as shown in TABLE 2. below. This goes further to confirm that our proposed model performs better than the other models.

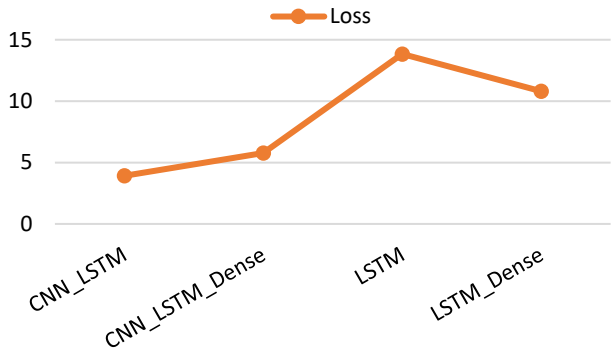

Fig. 4 Line graph showing loss of the different models on the iSPL dataset

Fig. 5 below illustrates the Softmax loss for the UCI dataset. Just like in Fig. 4 above, the CNN_LSTM model has the lowest loss value of 29.53% and is closely followed by the CNN_LSTM_Dense model at 30.18%. Unlike for the iSPL dataset above, the LSTM model shows a lower loss value compared to the LSTM_Dense model as shown in TABLE 2. further below. This can be attributed to the additional complexity of the model with an added layer. Model complexity is one of the key things that can explain an increase in the loss value.

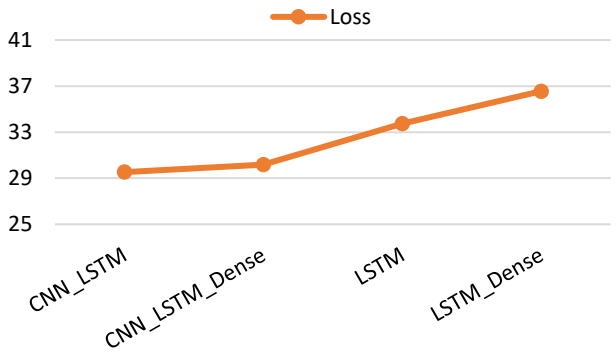

Fig. 5 Line graph showing loss of the different models on the UCI dataset

TABLE 2. SOFTMAX LOSS FOR THE CNN-LSTM MODEL ON BOTH THE ISPL AND UCI DATASETS

Model	iSPL	UCI
CNN_LSTM	**3.92%**	**29.53%**
CNN_LSTM_Dense	5.77%	30.18%
LSTM	13.84 %	33.73%
LSTM_Dense	10.81%	36.56%

The results above, to a larger extent, indicate that the proposed CNN_LSTM model outperforms the other deep learning architectures with a significantly higher accuracy rate while giving a low error rate compared to the other models. Adding a Dense layer reduced the accuracy slightly while increasing the loss value. The vanilla LSTM model gave mixed loss results but overall, it performed quite poorly when compared with the other 3 models.

V. CONCLUSIONS

In this paper, we proposed a CNN-LSTM approach to human activity recognition that seeks to improve the accuracy of activity recognition by leveraging the robustness in feature extraction of a CNN network while taking advantage of the work an LSTM model does for time series forecasting and classification. This CNN-LSTM model is both spatially and temporally deep and achieved better performance when it was compared with other deep learning approaches that use raw signal data as input. We evaluated this model regarding predictive accuracy and Softmax loss on both an internal (iSPL) and a publicly available (UCI HAR) dataset. In both cases, it outperformed the other models especially on the iSPL dataset with over 1% more accuracy that its closest rival and close to 2% less Softmax loss. Another metric that was not evaluated in this paper but was evidenced in the experiments was the time it took to run the different models and make predictions. It took much more time for the other models when compared to our proposed approach.

For future work, we shall develop this model further and properly evaluate it with different hyper parameters including the learning rate, batch size, regularization, and others. We plan to apply this model to more complex activities to tackle other challenges in deep learning and HAR by evaluating it on other datasets. We shall also evaluate this approach against the state-of-the-art results for the UCI dataset and other publicly available datasets.

ACKNOWLEDGMENT

This study was supported by the BK21 Plus project funded by the Ministry of Education, Korea (21A20131600011) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

REFERENCES

[1] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:*1604.08880 (2016).

[2] F. Chollet, "Home," *Keras Documentation*, 2015, Online, Available: https://keras.io/, Accessed: Dec. 1, 2019.

[3] R. Mutegeki and D. S. Han, "Feature-Representation Transfer Learning for Human Activity Recognition," *The 10th International Conference on ICT Convergence*, unpublished.

[4] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," *IEEE Int. Conf. on Big Data*, pp 1367–1376, 2018.

[5] A. Jain and V. Kanhangad, "Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors," *in IEEE Sensors Journal,* vol. 18, no. 3, pp. 1169-1177, 1 Feb.1, 2018. doi: 10.1109/JSEN.2017.2782492

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," *Proc. of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, 24–26 April, 2013.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, 25(2), pp. 1097–110, 2012.

[8] S. O. Eyobu and D. S. Han, "Feature Representation and Data Augmentation for Human Activity Classification Based on Wearable IMU Sensor Data Using a Deep LSTM Neural Network," *Sensors*, 2018, 18, 2892.

[9] F. M. Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst and M. Hompel, "Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors," *Informatics,* 5(2), 26, May 2018.

[10] K. Kim, S. Choi, M. Chae, H. Park, J. Lee, and J. Park, "A Deep Learning Based Approach to Recognizing Accompanying Status of Smartphone Users Using Multimodal Data," *Journal of Intelligence and Information Systems*, vol. 25, no. 1, pp. 163–177, Mar. 2019.

[11] F. Chollet, "Layer wrappers," *Keras Documentation*, 2015, Online, Available: https://keras.io/layers/wrappers/#timedistributed, Accessed: Dec. 1, 2019.

[12] S. Hochreiter and J. Schmidhuber. "Long short-term memory," *Neural computation*, 9(8):1735–1780, 1997.

[13] R. Gómez, "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names," Online, Available https://gombru.github.io/2018/05/23/cross_entropy_loss/, Accessed: Dec. 1, 2019.