EJECE, European Journal of Electrical Engineering and Computer Science
ISSN: 2736-5751

Improving the Performance of Closet-Set Classification in Human Activity Recognition by Applying a Residual Neural Network Architecture

Daniel Rodriguez Gonzalez and Yanzhen Qu

***Abstract* — The Time Series Classification Residual Network (TSCResNet) deep learning model is introduced in this study to improve the classification performance in the human activity recognition (HAR) problem. Specifically in the context of closed-set classification where all labels or classes are present during the model training phase. This contrasts with open-set classification where new, unseen activities are introduced to the HAR system after the training phase. The proposed TSCResNet model is evaluated with the benchmark PAMAP2 Physical Activity Monitoring Dataset. By using the same quantitative methods and data preprocessing protocols as previous research in the field of closed-set HAR, results show that the TSCResNet model architecture is able to achieve improved classification results across accuracy, the weighted F1-score, and mean F1-score.**

***Keywords* — Closed-set classification, deep learning, human activity recognition, residual neural networks.**

I. Introduction

Time series classification (TSC) involves the acquisition of time dependent data points and determining, via the use of machine learning algorithms, if the dataset belongs to a particular class of known patterns or types of observations [1]. For example, in an assembly line scenario, repetitive motions are common and can be easily predicted. Even in the case where there are several types of motions, time series classification models can accurately predict the type of motion observed. In closed-set classification, the training data contains all possible class labels that the machine learning algorithms can use to learn relationships between the training data and the class label of the data. Therefore, when the trained model is deployed to make predictions, the model will only predict classes present during the model training phase. The closed-set classification system is also expected to receive relevant data inputs for prediction or no new or unseen classes. This contrasts with the open-set classification case, new, previously unseen activities appear to a human activity recognition (HAR) system and so different frameworks are needed to assess classification performance [2].

This study focuses on the closed-set classification problem in the field of HAR. A new residual deep neural network architecture, TSCResNet, is proposed in this study with the aim to improve closed-set classification performance. The proposed method will be evaluated with an open-source dataset, the PAMAP2 Physical Activity Monitoring Dataset that is made available by the University of California Riverside (UCR) time series classification and clustering repository [3], [4]. By evaluating the performance of the proposed TSCResNet model with a commonly used standardized dataset in HAR research, it is possible to assess the effectiveness of this approach versus other methods.

II. Related Works

A. Deep Learning Architectures

In a recent publication by Lv *et al*. [2] deep learning based models are proposed to address the closed-set classification problem. An overview of the architectures, as illustrated by Lv *et al*. are shown in Fig. 1 [2].The models consists of five main components, a set of convolutional neural network (CNN) layers, a long short-term memory (LSTM) neural network, a pooling layer, a dense layer, and a custom modification to a softmax layer, or normalized exponential function, that adds a customized margin-based method to improve the classification performance and is therefore referred to as the arcmargin layer. These components are layered or connected sequentially to create the overall architecture of their model. This study focuses on two of the proposed architectures by Lv *et al*. [2]. One architecture combines all the components to create the Hybrid-M architecture, as shown on the bottom left of Fig. 1, and another only excludes the LSTM layers to create the CNN-M architecture, as shown in the top left of Fig. 1. The -M denotes their incorporation of the arcmargin layer. Lv *et al*. uses the open-source PAMAP2 Physical Activity Monitoring Dataset to present their findings and benchmark their deep learning architectures [2].

Similarly, Guan & Plötz and Moya Rueda *et al*. propose deep learning architectures to improve the closed-set classification performance using the PAMAP2 Physical Activity Monitoring Dataset [5], [6]. Guan & Plötz propose and ensemble method that consists of long short-term memory (LSTM) neural networks. LSTMs are a type of recurrent neural networks (RNN) that can account for sequential relationships in the data. Guan & Plötz also included the extra challenge in their research of addressing real-life scenarios where missing or erroneous data may be present [5].

Submitted on March 23, 2022.
Published on April 15, 2022.
D. Rodriguez Gonzalez, Colorado Technical University, USA.
(e-mail: danrgonzalez@gmail.com)
Y. Qu, Colorado Technical University, USA.
(corresponding e-mail: yqu@coloradotech.edu)

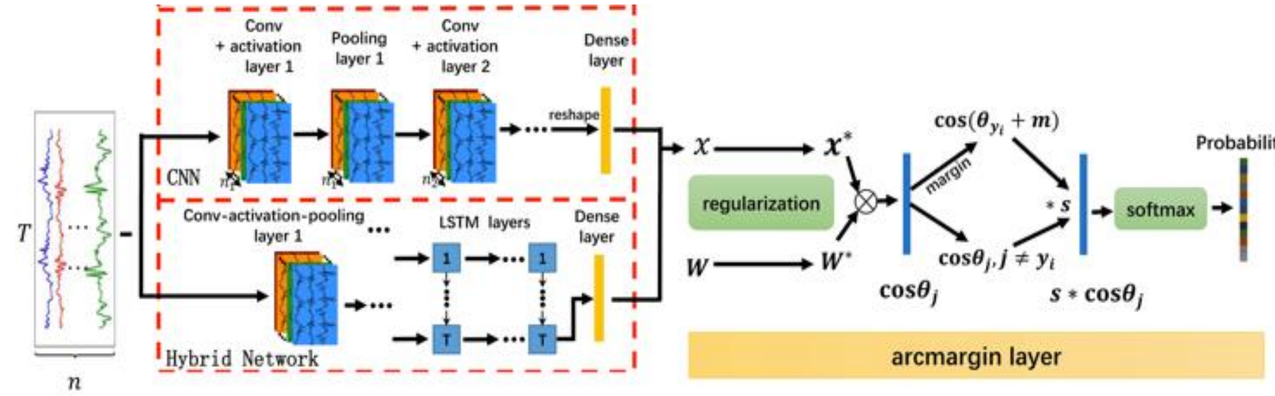

Fig. 1. Deep Learning Architectures by Lv *et al.* [2].

Moya Rueda *et al*. introduce the CNN-IMU network which follows the idea of wider rather than deeper neural networks [6]. The CNN-IMU architecture consists of parallel input branches that process each of the inertial measurement units (IMU) sensor data. Each branch consists of CNN-based temporal convolutions, subsequent pooling operations, and an additional fully connected layer [6]. For comparison, Moya Rueda *et al*. also implement a baseline CNN architecture that resembles that of a single branch CNN-IMU [6].

B. Deep Learning in Time Series Classification

The ideas behind using artificial neural networks and deep learning methods to tackle time series classification problems originates from two observations or characteristics of deep learning techniques. First, deep learning techniques can automatically learn high-level representations of raw input or training data. This property removes the manual approach to feature engineering, which is limited to an individual's ability to engineer features, and instead offers a method where the feature engineering process is fully automated and offers a more generic solution across classification problems [7]. Second, deep learning techniques have been applied across a variety of disciplines like computer vision, natural language processing, and speech recognition and have been able to achieve exceptional results [7].

In traditional sensor-based human activity recognition approaches, the feature extraction process involves hand-crafting features to develop a classifier that can recognize different types of activities. The drawback of these traditional methods is that they rely heavily on human experience or domain knowledge [2]. In recent years, with the rapid development of deep learning technology, the classification performance of human activity recognition based on deep learning networks has increased substantially [8].

Comparison studies have been conducted to assess the effectiveness of deep learning feature extraction techniques versus hand-crafted techniques [9]. Results show the effectiveness of deep learning methods over hand-crafted approaches. Similar results are shown in studies involving human forearm motion sensing, where a generalized model for movement recognition is introduced that is based on deep learning techniques is compared to traditional methods [10].

In contrast to previous traditional approaches, another researcher asserts that the previous efforts to adopt handcrafted features and design prediction models for each case, result in low accuracy due to ineffective feature representation and the limited training data [11]. Similarly, another group of researchers emphasize the same observation of artificial neural networks by stating that, most importantly, artificial neural networks help overcome several feature engineering issues, particularly in the manual definition of feature extraction procedures that are often erroneous or poorly generalizable. Therefore, the biggest advantage of contemporary deep learning methods is their ability to simultaneously learn both proper data representations and classifiers [5].

C. Convolutional Neural Networks

One class of Artificial Neural Network architecture that has gained recent attention in the field of time series classification is the use of deep neural networks with convolutional neural networks (CNN). In many cases, researchers precisely cite the performance of convolutional neural networks in other fields as the motivation for pursuing their research with convolutional neural networks. For example, one researcher states that in view of the superiority of convolutional neural networks in the field of computer vision, the same concept is applied to the classification of five mixed gas time series data collected by an array of eight MOX gas sensors [12].

Several studies have shown improvements in time series classification when employing deep convolutional neural networks. In one study, deep convolutional neural networks ingest different types of signals from body worn sensors like electrocardiograms, magnetometers and gyroscopes to predict 13 different types of activities with improved results [13]. In 2015, using deep convolutional neural networks, the proposal for an automated feature learning method to extract features for human activity recognition (HAR) tasks is presented [14]. Yang *et al*. [14] demonstrates that the deep convolutional neural network architecture can outperform other HAR algorithms through its automatic feature learning and classification process. This method is evaluated using the opportunity and other benchmark datasets [14].

Finally, in another more extensive study looking to evaluate the performance of multiple convolutional neural network architectures for human activity recognition, different performance levels can be achieved by the different architectures when training parameters like the learning rate or the max-pooling layers are modified [6].

D. Residual Neural Networks

A residual neural network is a feedforward neural network with "shortcut connections" that skip one or more layers [15].

The shortcut connection in effect performs identity mapping where the outputs are added to the outputs of the stacked layers. The shortcut connections to not add extra parameters or computational complexity. The entire network can still be trained with Stochastic Gradient Descent (SGD) with backpropagation using common deep learning libraries [15]. He *et al*. [15] demonstrated that ResNet architectures provide better network optimization and increased performance with greater layer depths [15]. Although the research by He *et al*. [15] primarily focuses on image classification tasks, the concept of the residual neural network can be extended to other fields.

E. Closed-Set Classification Evaluation Metrics

Time Series Classification models are optimized, compared, and evaluated with metrics, like classification accuracy, to assess the amount of error that the model's predictions generate. Lines and Bagnall conclude that classification accuracy is the most important performance metric in time series classification tasks [16]. Classification accuracy is therefore one of the main performance metrics that will be used in this study to evaluate model performance. However, datasets in the field of human activity recognition are known to be unbalanced. That is, one or two classes in the dataset represent a disproportionate amount of the total data. Therefore, the overall classification accuracy is not a suitable metric for measuring predictive performance [2]. To overcome this issue, a weighted F1-score can be used, which takes into consideration the precision and recall for each activity. A mean F1-score, which is independent of the class distribution in the dataset can also be used. In summary, Lv *et al*. [2] utilizes all three metrics to evaluate model performance against the PAMAP2 Physical Activity Monitoring Dataset, the classification accuracy, the weighted F1-score, and the mean F1-score. Since this study is using the same dataset, the same metrics will be computed to evaluate and compare the model performance to the results by Lv *et al*. [2], Guan & Plötz [5], and Moya Rueda *et al*. [6].

III. Problem, Hypothesis, and Research Question

A. Problem Statement

The classification performance as measured by the weighted F1-score, mean F1-score, and accuracy can be improved in the closed-set classification case in the context of human activity recognition by introducing residual neural network layers that are known to gain increased performance with greater neural network depths.

B. Hypothesis Statement

The proposed deep learning model with the TSCResNet architecture can improve the classification accuracy, weighted F1-score, and mean F1-score in the closed-set classification problem in the context of human activity recognition.

C. Research Question

Can the use of deep learning techniques, particularly with the implementation of a residual network architecture and convolutional neural networks, in the field of human activity recognition lead to improved predictive performance when dealing with a closed-set classification task in the context of human activity recognition?

IV. Methodology

A. Method

In this study, an alternative deep learning model will be proposed to address the closed-set classification problem in the context of human activity recognition. Unlike the model proposed by Lv *et al*. [2], this deep learning model architecture will not be fully sequential. Instead, the convolutional neural network (CNN) blocks will also be connected by utilizing skip connections or shortcuts to jump over some layers. This is a pattern known as a residual neural network (ResNet), which has been used in image classification tasks but has not been used in the context of human activity recognition (HAR). Additionally, the model proposed in this study will not utilize the arcmargin layer and will instead rely on the traditional softmax layer to compute the prediction probabilities. The same PAMAP2 Physical Activity Monitoring Dataset will be used to evaluate the model's performance and compare it to the results by Lv *et al*. [2], Guan & Plötz [5], and Moya Rueda *et al*. [6]. With the ResNet-based deep learning method proposed in this study, it is expected that the classification model will improve its performance and therefore provide a better model suited for closed-set classification in the field of human activity recognition.

The PAMAP2 Physical Activity Monitoring Dataset was introduced in 2012 by researchers in the field of physical activity monitoring and activity classification [4]. The dataset consists of 18 types of activities performed by 9 subjects wearing 3 inertial measurement units (IMUs) and a heart rate monitor. The IMUs are placed on the participant's dominant hand, chest, and ankle. Each activity is recorded for as much as three minutes. The IMUs captured time series data at a rate of 100 samples per second and the heart rate monitor operated at a sampling frequency of approximately 9 samples per second [4]. Each of the IMUs captured timestamped temperature data along with 3-dimensional data from gyroscopes, accelerometers, and a magnetometer. The types of activities performed by the subjects are lying, sitting, standing, ironing, vacuum cleaning, ascending and descending stairs, walking, Nordic walking, cycling, running and rope jumping.

As performed in other studies involving the PAMAP2 Physical Activity Monitoring Dataset, runs 1 and 2 for participant 5 will be used for validation, and runs 1 and 2 from the sixth participant will be used as the test split [4]. The remaining data will be used for training, which encompasses approximately 473,000 samples [5]. The performance evaluation of the proposed TSCResNet model follows the protocols used in previous research. As expressed by Guan and Plötz , unconstrained n-fold cross-validation is avoided [5]. Instead, fixed hold-out validation and test portions of the dataset are separated from the entire dataset. The remaining data is used as the training dataset. After each epoch during training, the model performance is evaluated with the validation set and the model with the best validation-set performance is then applied to the test dataset for the final set of results. The model performance is evaluated via 3 metrics:

the overall classification accuracy (Acc), the mean F1-score (Fm), and the weighted F1-score (Fw).

The computing platform is an 8-core Apple M1 processor with 16GB of memory. The TSCResNet model is implemented using the PyTorch library in the python programming language. The model will be developed in a conda virtual environment where all dependencies will be managed and the MLflow model tracking library will be used to track all training runs.

B. TSCResNet Architecture

The Time Series Classification Residual Network architecture, or TSCResNet, is proposed in this study as an alternative approach to improve the closed-set classification performance in the context of human activity recognition. As the name suggests, the TSCResNet model incorporates residual network components to address time series classification problems. Unlike the architectures proposed by Lv *et al*. [2], the TSCResNet architecture will not be fully sequential (see Fig. 1 and 2). Instead, the convolutional neural network (CNN) layers will also be connected by utilizing skip connections or shortcuts to jump over sequential CNN layers. This is a pattern known as a residual neural network (ResNet), which has been used in computer vision models for object detection and image classification but has not been used in the context of human activity recognition.

The TSCResNet architecture consists of three main blocks or layers. A CNNBlock, a ResidualBlock, and a PredictionBlock. The CNNBlock implements a 1-dimensional convolution over an input signal composed of several input planes. The CNNBlock is re-used multiple times through the entire network and at each implementation, the number of input and output channels can be defined. In a single forward pass through the block, the outputs of the 1-dimensional convolution layer are passed through a 1-dimensional batch normalization layer to reduce internal covariance shift and the activation function is defined to be a leaky rectified linear unit or LeakyReLU with a negative slope of 0.1. Batch normalization and a ReLU activation function is also used by Lv *et al*. [2].

The ResidualBlock component uses multiple CNNBlocks in its architecture. The ResidualBlock is defined with a value to indicate the number of repetitions that the ResidualBlock must build. For 1 repetition, two sequential CNNBlocks are linked together. It is in the forward pass through this single repetition that the residual nature of this layer is defined. When input data is passed through 1 repetition, the input data is temporarily saved while an additional copy is passed through the two sequential CNNBlocks. Once the output from the sequential CNNBlocks returns, it is added to the original input data. This process is repeated for each repetition that is defined in the creation of the ResidualBlock component. Fig. 2 illustrates the ResidualBlock architecture described here.

Finally, the PredictionBlock also consists of a pair of sequential CNNBlocks, but in the forward pass through the layer, the outputs of the sequential CNNBlocks are shaped into an 1-dimensional tensor that represents the predictions for the 18 different classes in the dataset. From these predictions, the index of the maximum value or argmax is identified along the 1×18 dimensional tensor. The index represents a particular class from the PAMAP2 Physical Activity Monitoring Dataset. Fig. 3 below shows a high-level overview of the entire TSCResNet architecture.

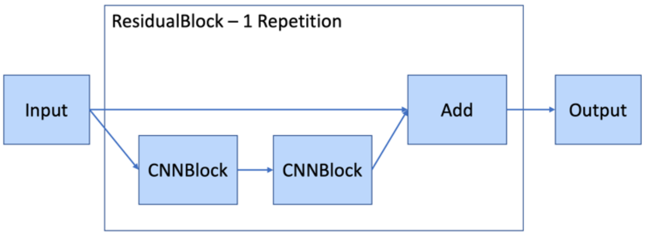

Fig. 2. A ResidualBlock Architecture for 1 Repetition.

Additionally, the model proposed in this study will not utilize additional customized layers like the arcmargin layer that Lv *et al*. introduce to help improve classification results on top of the base CNN-M model that they use [2]. Additionally, no max-pooling layers or dense layers are part of the TSCResNet architecture. Only a softmax function is used to transform the model's outputs to the predicted class probabilities.

V. EXPERIMENT AND RESULTS

A. Experiment Procedures

The purpose of this study is to improve the classification performance in the closed-set classification problem in the field of human activity recognition by utilizing ResNet-based deep learning techniques. This study will follow an experimental design, where the dependent variables are the classification performance metrics, accuracy (acc), mean F1-score (Fm) and weighted F1-score (Fw), and the independent variables are the classification models. The models differ in the deep learning architectures that they employ to achieve their predictive performance. The classification performance of the deep learning model proposed in this study will be compared to the results by Lv *et al*. [2], Guan & Plötz [5], and Moya Rueda *et al*. [6].

The model proposed in this study will be trained and evaluated with the PAMAP2 Physical Activity Monitoring Dataset. The PAMAP2 Physical Activity Monitoring Dataset consists of physical activity data collected by various sensors attached to individuals. This study will follow the same train, test, and validation splits as in other studies where runs 1 and 2 for participant 5 will be used for validation and runs 1 and 2 from the sixth participant will be used as the test split [2]. The remaining data will be used for training, which encompasses approximately 473,000 samples [5]. All model parameters will be randomly initialized at the beginning of each training run. The batch size will be set to 16 samples with an initial learning rate of 0.0001 and will be trained for 50 epochs.

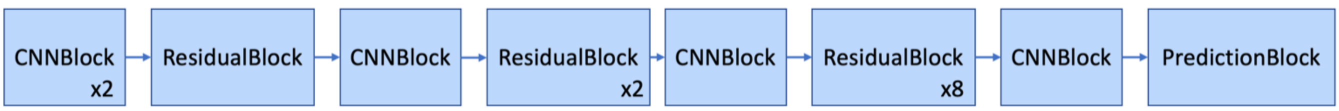

Fig. 3. High-Level Overview of the TSCResNet Architecture.

The model architecture and implementation will be developed using PyTorch, an open-source machine learning framework in the python programming language. Lv *et al*. [2] also implements their models using the pyTorch library with similar training hyperparameter settings.

Several data pre-processing steps are taken to follow the same experimental protocols used by other researchers. First, the PAMAP2 sensor data is down-sampled to 33.3 Hz to match the temporal resolution of other datasets used in similar studies. Down-sampling has been used by Guan and Plötz [5] and Lv *et al*. [2]. Additionally, Reiss & Stricker recommend using linear interpolation to deal with wireless data loss present in the IMU data [4]. Therefore, linear interpolation is also used to fill missing IMU values and missing values in the heart rate data. In the case that values are missing from the beginning of the sensor recordings, backfilling is used.

As indicated by Reiss & Stricker, data labeled with activity label 0 or other (transient activities) should be discarded in any kind of analysis [4]. This data mainly covers transient activities between performing different activities e.g. going from one location to the next activity's location or waiting for the preparation of some equipment. Similarly, IMU orientation data is specified to be invalid in the readme file provided by Reiss & Stricker and therefore columns 14-17 from each of the IMU sensors is discarded [4]. This results in 40 time series signals that can be used for training. Additionally, Reiss & Stricker delete 10 seconds from the beginning and the end of each labeled activity to avoid mixing with transient activity data [4]. These three data pre-processing steps are implemented in this study.

The training, validation, and test datasets are created by slicing the time-series data using sliding windows of 5.12 seconds with 78% overlap between adjacent windows [2]. Reiss & Stricker describe this segmentation as using a 5.12 second sliding window size, shifted by 1 second between consecutive windows [4]. This configuration yields approximately 14,000, 2,000, and 2,000 time-series data frames for the training, validation, and testing sets, respectively [2].

For each sliding window, its corresponding activity label is one-hot encoded into a 1-dimensional vector of size 18 as there are 18 total activities that are captured by the PAMAP2 Physical Activity Monitoring dataset. The index of the vector represents the activity if at a particular index, the value is 1. With one-hot encoding, all other positions in the vector are therefore 0.

For model training purposes, the final train, validation and test datasets are reshaped into 3-dimentional tensors before being loaded into the training environment. In this study, the training, validation, and test datasets are shaped into tensors of dimension (N, 40, 172). Where N is the number of windows being used for either the training, validation or test datasets. 40 is the number of signals being passed to the model (i.e. temperature, hand, ankle, & chest IMU data) and the 172 represents the number of timesteps per window to achieve the 5.12 second window size. The targets or activity labels to which these tensors correspond to are of tensor shape (N, 18). As described earlier, the target variable or activity labels are represented by a 1-dimensional vector of 18; 1 for each activity label in the dataset. As an example, a tensor of (100, 40, 172) for the training data, represents 100 windows consisting of 40 IMU signals with 172 timesteps. These 100 windows correspond to a tensor of (100, 18) to denote the activity label for each window.

The data pre-processing stage in this study resulted in the tensors for model training and evaluation as shown in Table I. This data is saved into a local directory where it is then retrieved by the training environment for a model training run.

TABLE I: TRAINING, VALIDATION, & TEST DATASET SHAPES

X_train:	(12719, 40, 172)	y_train:	(12719, 18)
X_val:	(2457, 40, 172)	y_val:	(2457, 18)
X_test:	(2252, 40, 172)	y_test:	(2252, 18)

During model training, several parameters and tracking metrics are setup. The open-source MLflow library and user interface is utilized as the tracking component for this study. The MLflow tracking component is an API and UI for logging parameters, code versions, metrics, and output files when running machine learning code and to proactively visualize and monitor the results. Each training run uses a python config module to load the model hyperparameters and settings. The most important parameters are the training batch size, number of epochs, the learning rate, and weight decay. Once these parameters are loaded into the training environment, they tracked with MLflow. Similarly, the most relevant metrics tracked by MLFlow are the model's loss per training epoch and the resulting validation and test accuracies, mean F1-scores and weighted F1-score after each training epoch.

Once the tracking metrics are setup, the TSCResNet model is loaded along with the model optimizer and loss function definitions. In this study, the pyTorch Adam optimizer is used and initialized with the learning rate and weight decay hyperparameters as defined in the python config module. The pyTorch CrossEntropyLoss is utilized as the loss function. The training data is randomly shuffled and passed through the network by the specified batch size.

B. Closed-Set Results

In Table II, the performance achieved by various deep learning architectures published in recent years by researchers in the field of human activity recognition that use the PAMAP2 dataset as their benchmarking dataset is compared. As shown in Table II, the TSCResNet architecture outperforms previous models across all three evaluation metrics. Achieving 95.6%, 95.49%, and 95.5% on the acc, Fw, and Fm metrics, respectively. Prior to the TSCResNet results, the best closed set classification accuracy achieved is 93.74% by Lv *et al*. [2] using their CNN-M model architecture. The CNN-M architecture is mainly composed of convolutional neural networks (CNN) and a custom arcmargin layer. The CNN-M architecture also achieved the greatest Fw score of 93.75%. However, the Lv *et al*. [2] Hydrid-M model architecture achieved the greatest Fm score of 93.09%. The research by Guan & Plötz [5] and Moya Rueda *et al*. [6] achieved good performance as well but did not beat the values provided by Lv *et al*. [2]. The TSCResNet 95.6% classification accuracy is visualized in Fig. 3 via a confusion matrix showing the counts per predicted activity versus its true activity label on the test dataset.

TABLE II: CLOSED-SET HAR RESULTS COMPARISON

Study	Architecture	PAMAP2 Closed-Set Classification Performance		
		Acc	Fw	Fm
Lv et al. (2020)	CNN-M	93.74%	93.75%	92.95%
	Hybrid-M	93.52%	93.52%	93.09%
Moya Rueda et al. (2018)	CNN-2	92.55%	92.60%	--
	CNN-IMU-2	93.13%	93.21%	--
Guan & Plötz (2017)	LSTM Ensemble	--	--	85.40%
	TSCResNet	**95.60%**	**95.49%**	**95.50%**

C. *Summary*

The overall results demonstrate that the TSCResNet model architecture is able to improve upon the performance achieved by previous models in the field. As shown in Table II, there is a 1.86% improvement in accuracy, a 1.74% improvement in the weighted F1-score, and a 2.41% improvement in the mean F1-score over the next best models per metric.

VI. CONCLUSION

The results in Table II show that the proposed deep learning model with the TSCResNet architecture can improve the classification performance across accuracy, weighted F1-score, and mean F1-score in the closed-set classification problem in the context of human activity recognition.

The classification improvements can be largely attributed to the TSCResNet model architecture. The CNN-M model used by Lv *et al.* [2], see Fig. 1, consists of convolutional layers followed by non-linear activity and pooling layers. There are two main differences between the CNN-M and TSCResNet model architectures. First, the TSCResNet model does not utilize pooling layers, see Figure 3. As stated by Lv *et al.* [2], the pooling layer performs down sampling which produces translational invariance. Instead, the TSCResNet model continues layering convolutional neural networks, thereby adding depth to the architecture and avoids any downsampling of the data, see Fig. 3. The additional CNN layers in turn help the model learn and abstract features from the data with the additional convolutional kernels. Second, the TSCResNet model introduces shortcut connections into the architecture where the ResidualBlock shown in Fig. 2 adds the inputs to the ResidualBlock to the output of the sequential convolutional layers. This residual learning approach has been studied in computer vision research which has shown that these residual networks are easier to optimize and can gain accuracy from increased depth [15].

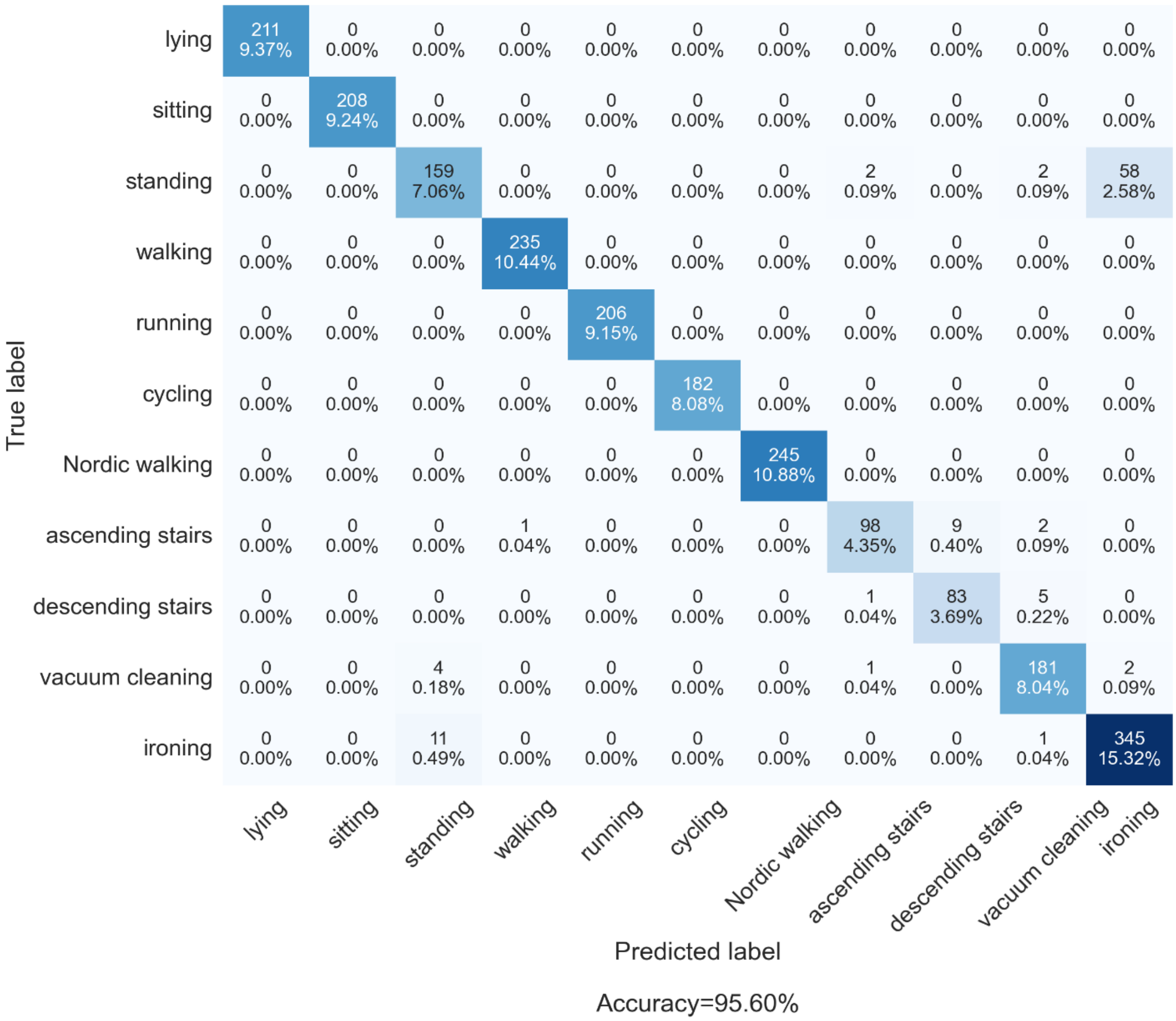

Fig. 4. Closet-Set Classification Confusion Matrix.

References

[1] Buza K. Time series classification and its applications. *8th International Conference on Web Intelligence, Mining and Semantics*; 2018 Jun 25-27; Novi Sad, Serbia.

[2] Lv T, Wang X, Jin, L, Xiao Y, Song, M. Margin-based deep learning networks for human activity recognition. *Sensors*. 2020;20(7). https://doi.org/10.3390/s20071871

[3] UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science [Internet]. [Retrieved 2021 May 28]. Available from: http://archive.ics.uci.edu/ml.

[4] Reiss A, Stricker D. Creating and benchmarking a new dataset for physical activity monitoring. *5th International Conference on PErvasive Technologies Related to Assistive Environments*; 2012; Crete Island, Greece.

[5] Guan Y, Plötz T. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2017;1(2):1-28. https://doi.org/10.1145/3090076.

[6] Moya Rueda F, Grzeszick R, Fink G, Feldhorst S, ten Hompel M. Convolutional neural networks for human activity recognition using body-worn sensors. *Informatics*. 2018; 5(2). https://doi.org/10.3390/informatics5020026.

[7] Saez Y, Baldominos A, Isasi P. A comparison study of classifier algorithms for cross-person physical qctivity recognition. *Sensors.* 2016;17(1). https://doi.org/10.3390/s17010066.

[8] Ordonez FJ, Roggen D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*. 2016;16(1). https://doi.org/10.3390/s16010115.

[9] Li F, Shirahama K, Nisar MA, Koping L, Grzegorzek M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors.* 2018;18(2). https://doi.org/10.3390/s18020679.

[10] Panwar M, Dyuthi SR, Prakash CK, Biswas D, Acharyya A, Maharatna K, *et al*. CNN based approach for activity recognition using a wrist-worn accelerometer. *Conf Proc IEEE Eng Med Biol Soc*; 2017, 2438-2441. https://doi.org/10.1109/EMBC.2017.8037349.

[11] Gu W, Zhou Z, Zhou Y, He M, Zou H, Zhang L. Predicting blood glucose dynamics with multi-time-series deep learning. *15th ACM Conference on Embedded Network Sensor Systems*. 2017;Delft, The Netherlands. https://doi.org/10.1145/3131672.3136965.

[12] Han L, Yu C, Xiao K, Zhao X. A new method of mixed gas identification based on a convolutional neural network for time series classification. *Sensors*. 2019;19(9). https://doi.org/10.3390/s19091960

[13] Uddin MZ, Hassan MM. Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network. *IEEE Sensors Journal*. 2019; 19(19): 8413-8419. https://doi.org/10.1109/jsen.2018.2871203.

[14] Yang JB, Nguyen MN, San PP, Li XL, Krishnaswamy S. Deep convolutional neural networks on multichannel time series for human activity recognition. *24th International Joint Conference on Artificial Intelligence*. 2015.

[15] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*. 2016: 770-778.

[16] Lines J, Bagnall A. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*. 2014; 29(3):565-592. https://doi.org/10.1007/s10618-014-0361-2.

Daniel Rodriguez Gonzalez received his B.S. in Engineering from Harvey Mudd College, US; Master's in Data Science from Southern Methodist University, US; and Doctorate in Computer Science from Colorado Technical University, US.

As a Machine Learning Engineer, he specializes in the development of Machine Learning and Deep Learning models using cloud technologies to train and deploy models at scale. He has worked on various complex projects where he has worked with business partners to define machine learning solutions, develop, and deploy the solutions to production environments. His current research interests include time series classification problems using deep learning and cloud computing.

Yanzhen Qu received his B.Eng. degree in Electronic Engineering from Anhui University, China; M.Eng. degree in Electrical Engineering from the Chinese Science Academy, China; and Ph.D. degree in Computer Science from Concordia University, Canada.

Over his industrial professional career, he has served at the various executive level management positions responsible for Product R&D and the IT Operation at several multinational corporations. He has led his multinational engineering teams successfully developed several the world first very large real-time commercial systems and technologies. He is currently the Dean and a Professor of Computer Science, Engineering and Technology with Colorado Technical University, Colorado Springs, CO, USA. He is also the dissertation supervisor of many Computer Science doctoral students. He and his doctoral students have published several dozen scholarly articles, some of them received the best paper award at several IEEE international conferences.

His current research interests include data science, cybersecurity and privacy, machine learning, e-learning technologies, software engineering, cloud computing, and affective computing. He has served as either the general chair, or the program chair, or a keynote speaker at many IEEE, ACM, ASIS, and IFIP international conferences or workshops. He is also an editorial board member of several professional peer-reviewed Computer Science or Information Technology journals.