Human activity recognition for packing processes using CNN-biLSTM

Alberto Angulo
alberto.angulo242400@potros.itson.edu.mx
Sonora Institute of Technology (ITSON)
Ciudad Obregon, Sonora, Mexico

Jessica Beltrán
jessicabeltran@uadec.edu.mx
Research Center in Applied Mathematics of the Universidad Autonoma de Coahuila (UAdeC)
Saltillo, Coahuila, Mexico

Luis A. Castro
luis.castro@acm.org
Sonora Institute of Technology (ITSON)
Ciudad Obregon, Sonora, Mexico

ABSTRACT

Human activity recognition has several applications in healthcare, sports, and industrial settings. In the latter, it can monitor industry workers and evaluate if the required activities are appropriately performed. In this paper, we employ state-of-the-art Deep Learning techniques to recognize ten distinct packing activities performed by sixteen participants from the Openpack dataset. Our proposed architecture combines data from various sensors and leverages Convolutional Neural Networks and Long Short-Term Memory networks to process spatial and temporal data. Additionally, we incorporate Transformers into our network, resulting in an improved F1-Score performance of 98.21.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

KEYWORDS

Human Activity Recognition, Packing Activities Recognition, Deep learning, Transformers

ACM Reference Format:
Alberto Angulo, Jessica Beltrán, and Luis A. Castro. 2023. Human activity recognition for packing processes using CNN-biLSTM. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing (UbiComp/ISWC '23 Adjunct), October 08–12, 2023, Cancun, Quintana Roo, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3594739.3610735

1 INTRODUCTION

Human activity recognition (HAR) research has progressed significantly in the last decade. Currently, HAR has successful applications, including healthcare [11], sports training [9], and human-computer interaction [18], among others. Accelerated advancement in sensor and computing technologies has driven the use of sensor data in HAR.

HAR has been applied in industrial environments to identify how employees perform activities, facilitating process improvement and optimization. Through HAR, it is possible to recognize how employees perform specific actions, making it possible to determine whether they are performed correctly, point out the mistakes they make, and generate recommendations for improving postures. It is also possible to identify actions related to workers' efficiency, such as correctly following packaging protocols and packaging anomalies.

Recently, research has been carried out trying to address this problem. A machine learning-based approach for multilevel recognition of industrial human activities is proposed in [16]. The system combines input from several state-of-the-art sensors, such as skeleton and fingertip poses. The authors used SVM and Random Forest to classify the activity and combined the result with a trained hierarchical hidden Markov model (HHMM). The results demonstrate the proposed approach's effectiveness in recognizing human actions in industrial environments, resulting in a 98% F-value performance.

Some existing challenges in HAR are data recognition using multimodal data, sequential activity recognition, and data recognition at different granularities. In product packaging, the problems are due to the heterogeneity in the size, the shape of the items to be packaged and the variability in how employees package the products.

This paper focuses on the use of HAR techniques for action recognition in the area of product packaging. A bidirectional deep convolutional network recognition model combining attention layers is proposed for classification. The Openpack dataset [22] is used to evaluate the proposed model. Finally, the proposed model is evaluated using the combinations provided by the authors of the Openpack dataset. Furthermore, to improve the robustness of our model, we have generated new training and validation sets while maintaining the same test set, a total of 3.

2 RELATED WORK

Human activity recognition is a challenging area of research that relies on sensors and has attracted attention in the deep learning fields. It is mainly based on sequential data. Evaluating and analyzing these signals is of special interest to achieve optimizations in the industry where manual labor is still dominant. Different methods have been developed to classify human movements. One example is the work in [15], where the authors recognized human activities performed in the assembly area by using a two-stage learning framework. The first stage used random forests and kernel-based support

UbiComp/ISWC '23 Adjunct, October 08–12, 2023, Cancun, Quintana Roo, Mexico

ACM ISBN 979-8-4007-0200-6/23/10…$15.00
https://doi.org/10.1145/3594739.3610735

vector machines and the second stage modeled the temporal dependencies between the resulting states with a hidden Markov model classification approach. The approach's performance was evaluated using exclusion cross-validation over a dataset of 24 recordings of workers performing activities in a human-robot interaction environment. The system achieved recognition accuracy of up to 88% for some activities and an average accuracy of 73%.

In addition, among traditional HAR methods are those based on Convolutional Neural Networks (CNN) where feature extraction is part of the neural network [13, 14]. A CNN that uses temporal convolutions applied over sequential data coming from inertial measurement units (IMU) was proposed in [4]. The CNN was designed to handle different sensor and IMU values separately, linking the information step by step within the architecture. The authors evaluated the architecture using order-picking process data recorded in two warehouses. The results show that the proposed architecture outperforms traditional approaches based on statistical features and recent CNN architectures. In addition, the effect of data augmentation is studied, and shows that networks trained with proper augmentation perform better than those trained without it.

For CNNs, extracting long-term dependency within sequential data is a difficult task. To solve this problem emerged the so-called long short-term memory (LSTM) networks [23], which can extract long-term dependencies within sequential data. A framework for activity recognition in surveillance videos captured in industrial systems is proposed in [19]. This can be applied to the automatic monitoring of hundreds of workers. The proposed framework combines MobileNet CNN and a multilayer LSTM network. The model was tested with different datasets such as UCTF101, HMDB51, HOLLYWOOD2, UCF50, and YOUTUBE. Overall, the proposed method outperforms benchmark architectures with up to 5% accuracy. In [21], a deep convolutional network combining CNN and LSTM was presented for HAR. The proposed approach was able to learn the temporal dynamics at various time scales increasing the accuracy.

In addition, LSTMs have been used with self-attention mechanisms that compute the correlation and weighted combination between all time steps in the input sequence [20]. The attention mechanism aims to learn the essential time steps from sequence feature maps that help determine more accurate recognition [5].

The transformer model [20] consists of multi-headed attention layers, fully connected layers, normalization layers, and dropout layers. In addition, it has residual connections that help with gradient backpropagation in a deep neural network. The transformer focuses directly on predicting the gain/loss intensity of the function during the feedforward phase as a function of the context encountered during the learning process. In [8], the transformer model is proposed for time series analysis of motion signals in human activity recognition. The self-attenuation mechanism expresses individual dependencies between signal values within a time series, which can match the performance of state-of-the-art CNNs with LSTM. It was tested using the largest publicly available sensor data (KU-HAR) dataset covering a wide range of activities, and an accuracy of 99.2% was obtained. While the results are good, the proposed method was only tested on a single dataset, and it is unclear how well it would perform on other datasets or in real-world scenarios.

In [3], a new weakly supervised human activity recognition model based on recurrent attention learning was proposed. The model was evaluated using the traditional UCI-HAR dataset and a weakly labeled collected set. Experimental results show that their model is superior to the conventional CNN and DeepConvLSTM models on both datasets.

3 PROPOSED APPROACH

We propose an activity recognition architecture using a fusion of convolutional networks with bidirectional LSTM. This approach allows us to extract time dependencies backward and forward, enabling us to predict the preceding and the following activities.

3.1 Dataset

We utilized the Openpack dataset [22] as it represents the most comprehensive dataset for human activity recognition in industry, specifically within the packaging sector. It is composed of sensor data and records from 16 participants performing packaging activities in industrial environments. These participants conducted various scenarios across different sessions. In Table 1 displays the scenario identifiers (ES01, ES02, ES03, ES04) pursued in each session, and in Table 2 elaborates on the scenario specifics. These scenarios are strikingly similar, with only slight variations such as allowing participants to alter the sequence of activities, the number of items, the alarm to simulate a work period, among other factors. Table 3 furnishes more intricate details about the Openpack dataset, such as the types of sensors used. Specifically for this study, data from the IMU accelerometer, key points from the Kinect sensor, bbox from the Kinect sensor, hand scanner, and printer were used, with no alteration in the frequencies.

Figure 1 displays the classes of packaging operations found in the Openpack dataset and indicates the number of minutes captured for each class. In total, ten sequential activities are evaluated. As it can be seen in Figure 1, the classes are unbalanced. The activities of relocating the item's label (i.e., operation ID = 2) and assembling the box (i.e., operation ID = 3) are the activities with the highest number of samples.

Table 1: Participants and sessions in OpenPack dataset

Participants		Sessions				
ID	**Dominant Hand**	**S0100**	**S0200**	**S0300**	**S0400**	**S0500**
U0101	R	ES01	ES01	ES01	ES01	ES01
U0102	R	ES01	ES01	ES01	ES01	ES01
U0103	R	ES01	ES01	ES01	ES01	ES01
U0105	R	ES01	ES01	ES01	ES01	ES01
U0106	R	ES01	ES01	ES01	ES01	ES01
U0107	R	ES01	ES01	ES01	ES01	ES01
U0109	L	ES01	ES01	ES01	ES01	ES01
U0111	R	ES01	ES01	ES01	ES01	ES01
U0202	R	ES02	ES02	ES03	ES03	ES04
U0205	R	ES02	ES02	ES03	ES03	ES04
U0210	R	ES02	ES02	ES03	ES03	ES04

Table 2: Details of scenarios in Openpack dataset

ID	Description
ES01	The participants followed the instructions as faithfully as possible. The list of items in an order was based on actual order sheets, but the variety of items contained in an order was limited to 54.
ES02	The participants were free to alter the operations procedures at their discretion. Also, the likelihood of including very large or small items in an order was reduced compared to ES01, and 21 new items were added.
ES03	Irregular situations/actions were introduced into ES02, such as pre-assembled shipping boxes that could be utilized by the workers, the inclusion of small items in paper bags, and the possibility of a subject carrying several consecutive orders of small items from the back table to the workbench simultaneously.
ES04	An auditory alarm was implemented in ES03 to simulate a busy work period, and periodic alarms were set (with an interval of 30-45 seconds) when the elapsed time of a period exceeded 80% of the average duration of a previously recorded work period.

Table 3: Openpack dataset details for recognition of human activities

Elements	Details
Type	Packaging Work Recognition
Participants	16
Sampling rate	IMU (Acc, Gyro, Ori): 30Hz Kinect (Keypoints, Bbox): 15Hz IoT sensors (Scanner, Printer): 1Hz
Activities (classes)	10 main + 32 secondary
No. of data obtained	20,129 work operations 52,529 shares
Modality	D+Keypoints+LiDAR+Acc+ Gyro+Ori+EDA+BVP+Temp
Recording duration	53h50m
Scenarios	4
Sessions	5 per participant

D: Depth, Acc: Accelerometer, Gyro: Gyroscope, EDA: Electrodermal Activity, BVP: Blood Volume Pulse, Temp: Temperature, Keypoints: Kinect sensor keypoints.

3.2 Data Preprocessing

Min-max normalization was applied to the input data using the maximum and minimum values detected in each sensor. Subsequently, the normalized data were rescaled to have a range from -1 to 1. Data synchronization was performed using Unix epochs since the data points were sampled at different frequencies.

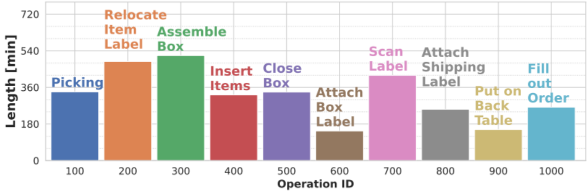

Figure 1: Total recording length of each operation [22].

3.3 Proposed Architecture

The network structure proposed in this work comprises separate sensor training that extracts spatial features for each sensor. Subsequently, a fusion of the features is performed, which is evaluated using a bidirectional LSTM network responsible for obtaining the temporal features. Finally, a feedforward layer is applied. The structure of the entire network Fusion CNN-BiLSTM can be seen in Figure 2, this structure is based on the architecture proposed by the *OpenpackChallenge* [1] winners.

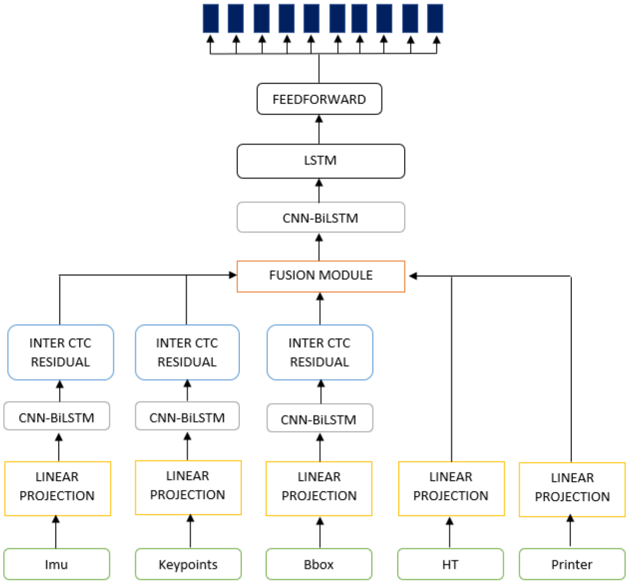

Figure 2: Model architecture for Fusion CNN-BiLSTM.

In the proposed architecture Fusion BiLSTM, there are two modules called CNN-BiLSTM and InnerLogits. The structure of the CNN-biLSTM is a variation of the architecture described in [12], this architecture was proposed in [1], combining convolutional, linear, and recurrent layers.

The CNN-biLSTM model [1] consists of 9 layers, it was configured with 100 filters in the convolutional layers and the batch normalization (BatchNorm) was changed to a Group Normalization (GroupNorm), this allows a consistent normalization regardless of the batch size and a GELU activation function. Bidirectional LSTM

[1] Openpack Challenge: https://open-pack.github.io/challenge2022, last access 24/07/2023

layers were established with 234 neurons with a dropout rate of 0.3. Fully connected linear layers were established with 330 and 223 neurons, respectively. Finally, the model output is given by a convolutional layer where the size is the model input.

Inspired by [3], Inter Connectionist Temporal Classification (CTC) [7] residual modules (Figure **??**) were added to make early predictions. During training and inference, each intermediate prediction is added to the input of the next layer to assist with recognition. The same method proposed in [3] was used, with a slight variation of CNN networks to enhance feature extraction. The intermediate CTC Loss predictions are stored and averaged to be used as feature input in the weighted combined loss function.

In addition to the proposed approach in Figure 2, we also propose a variation that consists of changing the CNN-BiLSTM module for the Transformer CNN-BiLSTM module. As shown in Figure 3, inspired by the winners of the *OpenpackChallenge* [2]. As shown in Figure 3, the module consists of a pre-normalization layer followed by an attention layer that provides us with context for any position in the time series. This mechanism is crucial for the transformer model, where each attention head in multiple attention heads can look for a different relevance definition or correlation [8], the attention layer consists of 20 heads with dimension 64, the nowcast is added to the input of the next layer to aid recognition. A pre-normalization is performed and followed by the CNN-biLSTM module is applied in the same way the nowcast is added to the input of the next layer. This process is performed 3 times, that is, with a depth of 3. We call this proposed network the CNN-BiLSTM Fusion Transformer, which can be seen in Figure 4.

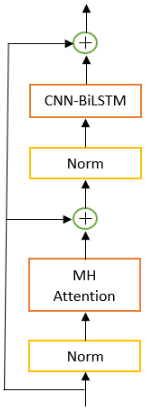

Figure 3: Model architecture for Transformer CNN-BiLSTM.

3.4 Model Training

The proposed architecture was implemented in Pytorch, and the weighted combined loss function was used for network optimization. Given a cross-entropy loss function $C(y, y')$, where y is the model output and y' is the true label, the weighted loss function L for a convolutional neural network model (CNN) and another feature model (F) is defined as:

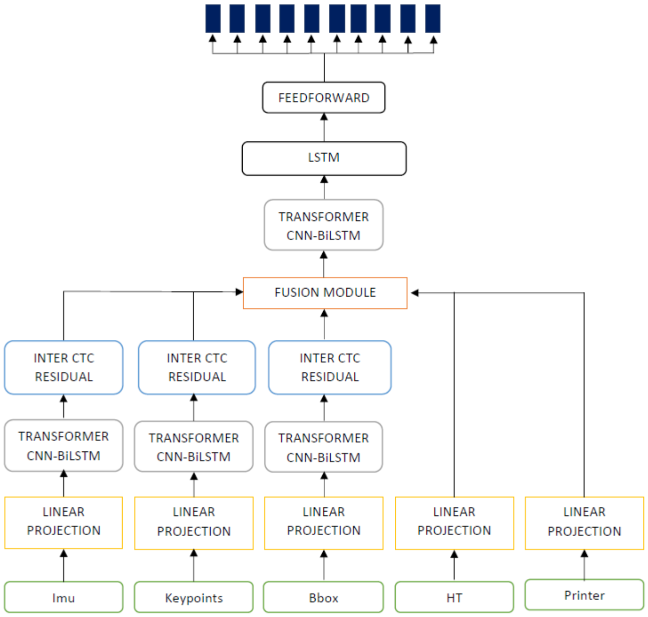

Figure 4: Model architecture for CNN-BiLSTM Fusion Transformer.

$$L = \alpha * C(CNN) + (1 - \alpha) * C(F) \tag{1}$$

where:

- α is a hyperparameter in the interval [0,1] that determines the weight of each loss in the total loss,
- $C(CNN) = C(yCNN, y')$ is the cross-entropy loss of the CNN model, and
- $C(F) = C(yF, y')$ is the cross-entropy loss of the feature model.

The input to the network consists of a data sequence formed by time series extracted from the sensor data Atr (Acc, Gyro, Ori), Kinect (Keypoints, Bbox). The data were segmented into batch sizes of 32 data per second to demonstrate the model's efficiency. We used the Adam optimizer [6], employing a warmup-based scheduler [20] to optimize the learning rate. This helps us avoid overly large or unstable gradients at the beginning of training, which can destabilize the learning process. By starting with a lower learning rate, the model can find a reasonable local minimum before exploring the parameter space with a more considerable step.

The model was trained over the training data composed of data coming from the participants and sessions described in Table 5.

The model was trained in Pytorch with the hyperparameters described in Table 4. Previous works [2, 10, 17] reported that using a learning rate of 0.0001 is sufficient when using the Adam Optimizer for human activity recognition. In addition, we use *WarmupScheduler* [20] to handle the learning rate during training. The purpose of *warmup* is to start with a low learning rate at the beginning of training and then gradually increase it to a predefined value. This approach helps us to prevent the network training from stagnating

[2] Openpack Challenge: https://open-pack.github.io/challenge2022, last access 24/07/2023

at the start and can assist the network in achieving better final accuracy.

3.5 Evaluation

The multiclass overall classification accuracy measure from the *Torchmetrics*[3] library was used to evaluate the architecture's performance during training. As this is an imbalanced dataset, the results could achieve high performance if the classifier predicts each instance as a majority class and overall classification accuracy is used to evaluate the outcome. Therefore, more appropriate measures exist to evaluate the model than overall classification accuracy. On the other hand, the F1-score considers both false positives and false negatives and shows the balance between precision and recall. Precision can be seen as $\frac{TP}{TP+FP}$ and recall as $\frac{TP}{TP+FN}$, where TP and FP are the number of true and false positives, respectively. The F1-score is given by equation 2:

$$F1 - score = \sum_{i=1}^{N} 2 * w_i \frac{precision_i.recall_i}{precision_i + recall_i} \tag{2}$$

Where $w_i = \frac{n_i}{N}$ is the proportion of the ith class sample, with n_i being the number of samples from the ith class, and N being the total number of samples.

To evaluate the effectiveness of our model, we adopted a two-stage data partitioning approach. First, we split the original dataset into the training (Tr), validation (V), and testing (Te) datasets described in Table 5.

Table 4: List of selected hyperparameters

Scenario	Hyperparameters	Selected values
Data processing	Window size	5000
	Optimizer	Adam
	Batch Size	32
Training	Learning Rate	0.0001
	Weight Decay	0.0001
	Epochs	200

Table 5: Configurations used to evaluate CNN-biLSTM architecture for Training (Tr), Validation (V), and Testing (Te).

	Participant	Session
Tr	U0101, U0102, U0103, U0105, U0106, U0107,U0109, U0110	S0100, S0200, S0400, S0500
V	U0101, U0103, U0105, U0107, U0109, U0111, U0205	S300
Te	U0102, U0106, U0202, U0210	S300

Second, to enhance the robustness of our model, we generated three randomized training and validation sets from the training and validation data described above, which we used to further train our model. We kept a separate test set for our final evaluation. This approach allows us to introduce greater variability into the training and validation data, which can help to improve the model's generalization capabilities and its robustness to new data, the second configuration described in Table 6.

Table 6: Configurations used to evaluate Transformer CNN-biLSTM architecture for Training (Tr) and Validation (V).

		Participant	Session
First	**Tr**	U0103, U0105	S0100, S0200, S0300, S0400, S0500
		U0107, U0109	S0100, S0200, S0300, S0400, S0500
		U0111, U0205	S0100, S0200, S0300, S0400, S0500
		U0106, U0202	S0100, S0200, S0400, S0500
		U0210	S0100, S0200, S0400, S0500
	V	U0101	S0100, S0200, S0300, S0400, S0500
		U0102	S0100, S0200, S0400, S0500
Second	**Tr**	U0101, U0103	S0100, S0200, S0300, S0400, S0500
		U0105, U0107	S0100, S0200, S0300, S0400, S0500
		U0111, U0205	S0100, S0200, S0300, S0400, S0500
		U0102, U0202	S0100, S0200, S0400, S0500
		U0210	S0100, S0200, S0400, S0500
	V	U0109	S0100, S0200, S0300, S0400, S0500
		U0106	S0100, S0200, S0400, S0500
Thrid	**Tr**	U0101, U0105	S0100, S0200, S0300, S0400, S0500
		U0107, U0109	S0100, S0200, S0300, S0400, S0500
		U0111, U0205	S0100, S0200, S0300, S0400, S0500
		U0102, U0106	S0100, S0200, S0400, S0500
		U0202	S0100, S0200, S0400, S0500
	V	U0103	S0100, S0200, S0300, S0400, S0500
		U0210	S0100, S0200, S0400, S0500

We adopted a 'performance-based model checkpointing' strategy. Specifically, after each training epoch, we evaluated the model's performance on the validation set. If this performance exceeds the best performance observed in previous epochs, we saved the model's current state as a 'checkpoint.' This approach allows us to retain the model that has demonstrated the best performance on the validation set rather than simply using the model's state at the final training epoch. These 'checkpoints' were used to evaluate the test set. The data used for model evaluation bear no relation to the Openpack Challenge, as until the submission date of this article, the annotation data used for this challenge are not public, and it is not possible to evaluate them.

4 RESULTS

To test the robustness of our model, we utilized the randomized datasets described in section 3.5 to evaluate over the same test set. This ensures that the model's final evaluation is performed on a dataset not seen during training or validation, providing an unbiased and robust assessment of the model's performance on new and previously unseen data.

Using the settings from Table 5, an accuracy of 96.20% was achieved. As can be seen, the Fusion Transformer CNN-BiLSTM outperforms Fusion CNN-biLSTM by 2.01%, achieving an accuracy

[3] Torchmetrics: https://torchmetrics.readthedocs.io/en/stable/classification/accuracy.html, last access 14/06/2023

of 98.21%. This suggests that using transformers in training our model improved the results.

In addition, a training was carried out using the code provided by the OpenPack Challenge winners, the results can be seen in Table 7. It can be seen that the applied variations exceed the model proposed by the Openpack Challenge winners in the 3 configurations, in addition to achieving an improvement of 1.76% on average.

Table 7: F1 scores for second experiment using configurations of Table 6.

Model	Type	F1 (%)
OpenPack Winner	First Conf	96.19
	Second Conf	96.57
	Third Conf	96.60
	Average	96.45
Transformer CNN-biLSTM	First Conf	98.47
	Second Conf	98.15
	Third Conf	98.03
	Average	98.21

5 CONCLUSIONS

Our work aims at classifying packing processes using the Openpack dataset. Our approaches, deemed Fusion CNN-BiLSTM and CNN-BiLSTM Fusion Transformer, yielded promising results. Our F-score was 96.20% for the Fusion CNN-BiLSTM and 98.21% CNN-BiLSTM Fusion Transformer.

In this study, we utilized fixed initial hyperparameters, as prior research demonstrated their effectiveness when dealing with human activity recognition. This is an ongoing work that has showcased the substantial potential of the proposed architecture, achieving a result of 98.21% even without hyperparameter optimization.

Our work could benefit from a larger dataset. We could also benefit from testing the proposed approach with different datasets to see whether the results can be generalized to similar problems.

ACKNOWLEDGMENTS

This work was funded by the National Council for Humanities, Sciences, and Technologies in Mexico (CONAHCYT) with a scholarship provided to the first author, and by the Instituto Tecnológico de Sonora through the PROFAPI program.

REFERENCES

[1] Alberto Angulo, Luis Castro, and Jessica Beltrán. en prensa. Reconocimiento de acciones de empaquetado usando redes CNN-biLSTM y optimización bayesiana. (en prensa).

[2] Hritam Basak, Rohit Kundu, Pawan Kumar Singh, Muhammad Fazal Ijaz, Marcin Woźniak, and Ram Sarkar. 2022. A union of deep learning and swarm-based optimization for 3D human action recognition. *Scientific Reports* 12 (12 2022). Issue 1. https://doi.org/10.1038/s41598-022-09293-8

[3] Maxime Burchi and Radu Timofte. [n. d.]. Audio-Visual Efficient Conformer for Robust Speech Recognition. https://github.com/burchim/AVEC.

[4] Rene Grzeszick, Jan Marius Lenk, Fernando Moya Rueda, Gernot A. Fink, Sascha Feldhorst, and Michael Ten Hompel. 2017. Deep neural network based human activity recognition for the order picking process. *ACM International Conference Proceeding Series* Part F131931. https://doi.org/10.1145/3134230.3134231

[5] Rebeen Ali Hamad, Masashi Kimura, Longzhi Yang, Wai Lok Woo, and Bo Wei. 2021. Dilated causal convolution with multi-head self attention for sensor human activity recognition. *Neural Computing and Applications* 33 (10 2021), 13705–13722. Issue 20. https://doi.org/10.1007/s00521-021-06007-5

[6] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015* (12 2014). https://doi.org/10.48550/arXiv.1412.6980

[7] Jaesong Lee and Shinji Watanabe. 2021. Intermediate Loss Regularization for CTC-based Speech Recognition. (2 2021). http://arxiv.org/abs/2102.03216

[8] Iveta Dirgová Luptáková, Martin Kubovčík, and Jiří Pospíchal. 2022. Wearable Sensor-Based Human Activity Recognition with Transformer Model. *Sensors* 22 (3 2022), 1911. Issue 5. https://doi.org/10.3390/s22051911

[9] Sakorn Mekruksavanich and Anuchit Jitpattanakul. 2022. Multimodal Wearable Sensing for Sport-Related Activity Recognition Using Deep Learning Networks. *Journal of Advances in Information Technology* 13 (4 2022), 132–138. Issue 2. https://doi.org/10.12720/jait.13.2.132-138

[10] Saeed Mohsen. 2023. Recognition of human activity using GRU deep learning algorithm. *Multimedia Tools and Applications* (2023). https://doi.org/10.1007/s11042-023-15571-y

[11] Godwin Ogbuabor and Robert La. 2018. Human Activity Recognition for Healthcare using Smartphones. *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, 41–46. https://doi.org/10.1145/3195106.3195157

[12] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors (Basel, Switzerland)* 16 (1 2016). Issue 1. https://doi.org/10.3390/s16010115

[13] Madhuri Panwar, S. Ram Dyuthi, K. Chandra Prakash, Dwaipayan Biswas, Amit Acharyya, Koushik Maharatna, Arvind Gautam, and Ganesh R. Naik. 2017. CNN based approach for activity recognition using a wrist-worn accelerometer. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2438–2441. https://doi.org/10.1109/EMBC.2017.8037349

[14] Md. Ashikur Rahman, Yousuf Mia, Mizanur Rahman Masum, Dm. Mehedi Hasan Abid, and Tariqul Islam. 2022. Real Time Human Activity Recognition from Accelerometer Data using Convolutional Neural Networks. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 1394–1397. https://doi.org/10.1109/ICCES54183.2022.9835797

[15] Alina Roitberg, Alexander Perzylo, Nikhil Somani, Manuel Giuliani, Markus Rickert, and Alois Knoll. 2014. Human activity recognition in the context of industrial human-robot interaction. *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 1–10. https://doi.org/10.1109/APSIPA.2014.7041588

[16] Alina Roitberg, Nikhil Somani, Alexander Perzylo, Markus Rickert, and Alois Knoll. 2015. Multimodal Human Activity Recognition for Industrial Manufacturing Processes in Robotic Workcells. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 259–266. https://doi.org/10.1145/2818346.2820738

[17] Jaka Septiadi, Budi Warsito, and Adi Wibowo. 2020. Human Activity Prediction using Long Short Term Memory. *E3S Web of Conferences* 202. https://doi.org/10.1051/e3sconf/202020215008

[18] Chamani Shiranthika, Nilantha Premakumara, Huei-Ling Chiu, Hooman Samani, Chathurangi Shyalika, and Chan-Yun Yang. 2020. Human Activity Recognition Using CNN & LSTM. *2020 5th International Conference on Information Technology Research (ICITR)*, 1–6. https://doi.org/10.1109/ICITR51448.2020.9310792

[19] Amin Ullah, Khan Muhammad, Javier Del Ser, Sung Wook Baik, and Victor Hugo C. de Albuquerque. 2019. Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM. *IEEE Transactions on Industrial Electronics* 66 (12 2019), 9692–9702. Issue 12. https://doi.org/10.1109/TIE.2018.2881943

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. (6 2017).

[21] Kun Xia, Jianguang Huang, and Hanyu Wang. 2020. LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access* 8 (2020), 56855–56866. https://doi.org/10.1109/ACCESS.2020.2982225

[22] Naoya Yoshimura, Jaime Morales, Takuya Maekawa, and Takahiro Hara. 2022. OpenPack: A Large-scale Dataset for Recognizing Packaging Works in IoT-enabled Logistic Environments. (12 2022). https://doi.org/10.48550/arXiv.2212.11152

[23] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation* 31 (7 2019), 1235–1270. Issue 7. https://doi.org/10.1162/neco_a_01199

Received June 2023; revised 12 March 2009; accepted 5 June 2009