Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2022, Article ID 1681096, 8 pages
https://doi.org/10.1155/2022/1681096

Research Article

LSTM-Based Neural Network to Recognize Human Activities Using Deep Learning Techniques

Sunitha Sabbu and Vithya Ganesan

Koneru Lakshmaiah Education Institute, Vaddeswaram, Andhra Pradesh, India

Correspondence should be addressed to Sunitha Sabbu; sunithacse79@gmail.com

Received 2 June 2022; Revised 10 July 2022; Accepted 14 July 2022; Published 23 August 2022

Academic Editor: Ridha Ejbali

Deep learning techniques have recently demonstrated their ability to be applied in any field, including image processing, natural language processing, speech recognition, and many other real-world problem-solving applications. Human activity recognition (HAR), on the other hand, has become a popular research topic due to its wide range of applications. The researchers began working on the new ideas by combining the two emerging areas to solve HAR problems using deep learning. Recurrent neural networks (RNNs) in deep learning (DL) provide higher opportunity in recognizing the abnormal behavior of humans to avoid any kind of security issues. The present study proposed a deep network architecture based on one of the techniques of deep learning named as residual bidirectional long-term memory (LSTM). The new network is capable of avoiding gradient vanishing in both temporal and spatial dimensions with a view to increase the rate of recognition. To understand the complexity of activities recognition and classification, two LSTM models, basic model and the proposed model, were used. Later, a comparative analysis is performed to understand the efficiencies of the models during the classification of five human activities like abuse, arrest, arson, assault, and fighting images classification. The basic LSTM model has achieved a training accuracy of just 18% and testing accuracy of 21% with higher training and classification loss values. But the proposed LSTM model has outperformed the basic model while achieving 100% classification accuracy. Finally, the observations have proved that the proposed LSTM model is best suitable in recognizing and classifying the human activities well even for real-time videos.

1. Introduction

Engineered features obtained through heuristic processes have traditionally been used to solve HAR tasks. Deep neural networks may be able to automate extraction of features from raw video inputs, according to current research findings. HAR, on the other hand, relies on capturing the temporal dynamics of human abnormal activities, which involves sequence of more complex activities. Because recurrent neural networks (RNNs) related to time series domains have recently been successful, we present an all-encompassing, general deep framework to recognize activity based on convolutional network and long short-term memory (LSTM) recurrent units. Our research focuses on the performance impact of key architectural hyperparameters, with the goal of identifying ways to improve those impacts.

Deep learning techniques appear to have the potential to meet the needs of activity recognition through videos. First and foremost, performance can be improved through the current proposed system. Second, deep learning (DL) techniques are able to identify features related to the dynamics of producing the human motion, ranging from lower-layer simple activity encoding to upper-layer more complex activity dynamics. This could be useful for increasing the complexity of activity recognition.

Video analysis can be viewed as a form of time series data modeling, which is similar to the HAR application discussed in this paper. Videos can be better classified by combining the temporal information in continuous and subsequent frames of a video with CNNs and LSTMs in the video domain. For the best results, LSTM cells were found to be essential in a comparative analysis of the Kaggle datasets, resulting in the highest reported performance.

The main contributions of the paper are as follows:

(i) We present deep convolutional network with LSTM recurrent layers that can learn the image features.

(ii) We demonstrate that the present approach is so efficient in recognizing abnormal activities from input videos by implementing to solve the problems of human activity recognition.

(iii) We exhibit that the proposed method works directly on the image frames extracted from the videos with minimal preprocessing.

(iv) We discuss the experimental process and summarize the results for further research taking the benefit of the deep learning architectures.

The large variability in human movements used for a given action makes human activity recognition difficult. To address increasingly complex recognition problems, we are motivated by two requirements like improving the recognition accuracy and at the same time reducing relay on engineered features.

The other sections of the paper are organized as follows: Section 2 describes the survey conducted to understand the nature of existing recognition systems. Section 3 describes the complete research methodology followed to conduct the data collection and feature extraction and address the classification process. Section 4 describes the experimental work and its outcomes followed by conclusion and references.

2. Literature Survey

Recent studies on using deep learning methods in HAR were based on deep belief networks (DBNs) [1] developed by holding multiple layers of Boltzmann Restricted Machine (RBM). Above the RBM layers and through the implementation of hidden Markov models (HMMs), subsequent DBN-based models exploited intrinsic temporal human activities sequences [2]. They performed a pretraining phase that was unsupervised to produce intrinsic features and used the available data labels to adjust the model. Nonetheless, HMMs are constrained by their number of possible hidden layers and have become inefficient in wide context windows when modeling long-range dependencies. The use of HAR-based convolutional neural networks (CNNs) was introduced [3] but they used only a single accelerometer and a shallow model.

The configuration [4] used only a single accelerometer for multisensor recognition with deep CNNs built [5] and later proposed using two accelerometers. A new proposed multichannel CNNs time series architecture was developed [6]. The architecture was a compact shallow model, and convolutional layers have been applied to the inertial signal spectral domain [7]. The proposed model was designed for devices with low power using a spectrogram of the input data; it reintroduced the extraction of handcrafted features. Successful implementation of convolutional networks for HAR is attributed to their ability to learn efficient and more discriminative features using 1D temporal convolutions in sequence to capture internal dependencies between adjacent input samples. CNNs use parameter sharing over time to capture local dependencies applying the same convolutional kernel at every time segment and local connectivity neurons getting feedback from small clusters of input samples among adjacent layers [8]. Sharing the parameters over time, however, is not enough to capture all the associations between input samples. Local synchronization often restricts output to a small number function of adjacent input data samples.

The researchers' proposed method utilizes a recurrent neural network to analyze the actions in videos. RNNs are designed to build blocks of a neuron that are connected to a central hub. They can then be used to process data in sequence [9]. The goal of this study was to recognize the actions of people in videos by analyzing the information from the previous frames. Through LSTM, the researchers were able to make an automatic assessment of the actions in videos [10]. The proposed method utilizes a recurrent neural network (LSTM) to analyze the actions in videos. RNNs are designed to build blocks of a neuron that are connected to a central hub. These blocks can be used to process data in sequence. They can also model outputs that are not independent of the central hub [11].

The researchers have presented various kinds of LSTM, such as multilayer and bidirectional LSTM. The researchers' proposed method is able to analyze the complex patterns in the visual data of each frame. It cannot be efficiently implemented with simple LSTM and multilayer LSTM [12]. The researchers' proposed method takes into account the features of video frames to recognize the actions in them. Deep features from each frame are then extracted using a pretrained AlexNet [13]. The two layers of the proposed architecture are then used to learn the sequence information in each frame. The proposed method is able to recognize actions in long videos because the data collected in the sequence is processed in N time steps.

3. Research Methodology

3.1. Data Collection. In this research, the problem of human activity recognition on the basis of movement-based features is addressed with six different activities like abuse, arrest, arson, assault, and fighting. The videos to generate datasets for each activity are collected from the Kaggle repository: four videos for abuse, one video for arrest, two videos for arson, one video for assault, and two videos for fighting. Later, using the proposed key frames extraction process, a total of 2250 frames were generated for training dataset with 450 frames for each activity class, and a total of 90 frames for testing dataset with number of frames for each activity class are used for further human activity recognition and classification process.

3.2. Extraction of Image Frames. In numerous applications, such as video categorization, activity detection, and video summarizing, detecting important frames in videos is a typical task. Instead of requiring the entire video, these activities can be accomplished with just a few essential frames. Existing detection methods based on key frames are primarily intended for supervised learning (SL) and require manual labeling in a large amount of training data in order to run the models. Labeling necessitates the use of human annotations from various annotated backgrounds with important frames in real-time videos; that is not costly and time-consuming but also leads to inconsistences and subjective errors among the labelers.

Videos typically have a frame rate of 30 frames for each second and provide additional information required to handle computer vision applications. Video summarization, activity identification, and visual localization and mapping are examples of key frame detection applications. Processing all frames necessitates a large amount of memory and computer power. In many situations, a few critical frames, or perhaps just one, may be enough to accomplish the needed outcomes; for example, some activities can be recognized from one single frame. Similarly, video summarizing is the process of identifying significant frames in a movie and using those frames to summarize the full video's information.

Allowing human participants to view a video and annotate significant frames is a common technique to solving this problem. It is impossible to get a consensus on the crucial frames due to the task's inherent subjectivity. The tagged key frame videos can be used to train models to automate the detection of key frame in previously viewed videos. Then, the annotated key frames are used as the gold standard in this scenario. Another option is to apply deep neural networks, which can provide cutting-edge results for a variety of visual tasks, including critical frame detection. Deep models, on the other hand, necessitate massive datasets for training that are time-consuming for humans to annotate.

The present research follows an autonomous self-driven approach for finding key video frames to address these issues. To detect frames that are unique, the suggested LSTM learns deep appearance and motion properties. After that, the trained network can recognize crucial frames in test videos. A total of 100 key frames per second were generated over an entire duration of the video. Figure 1 shows sample images related to five different human activities.

3.3. Data Preprocessing: Removal of Duplicates and Feature Extraction.

Duplicate images in your dataset are problematic for two reasons:

(i) They introduce bias into your dataset, providing additional opportunities for your deep neural network to learn patterns specific to the duplicates.

(ii) They impair your model's ability to generalize to new images other than those on which it was trained.

While we frequently assume that data points in a dataset are independent and identically distributed, this is rarely (if ever) the case when working with a real-world dataset.

The basic logic behind this process is to generate a hash value for each image based on its pixel value and count rather than its name. Based on this hash value, we will store the images in a dictionary with the hash value as the key and the binary value of the image as the value. Based on this, we store the images in a dictionary, or if we find a duplicate, we simply add it to a duplicate list with an index and image binary form. This list of images is later deleted based on their index value.

The process of transforming raw data into numerical features that can be processed to preserve the information in the original dataset is referred to as feature extraction. It produces better results than directly applying raw data to machine learning. Feature extraction can be done both manually and automatically. In this research, automatic feature extraction process named FET-3 is proposed. Feature extraction is carried out as a systematic and a step-by-step process in which three internal extraction techniques like grayscale pixel value (GCPV) features, mean values of channel pixels (MVCP) features, and edge extraction (EE) features were considered.

Outcome: Single channel (1D) grayscale image of length *nf*.

Using the raw pixel values as separate features is the simplest way to create features from a grayscale image with single color channel. During the extraction, an image size of 320×240 (width and height) is considered in which the number of features will be the same as the number of pixels. This means that each image with the above dimensions can produce 76800 pixels as features. During the process, every pixel value is appended to each other to generate a new feature vector. When the image is a 3-channel (RGB), the number of features in our case is $320 \times 240 \times 3$, that is, 2,30,400. The alternative for this approach is generating a new matrix by considering the mean value of pixels from all the 3 color channels. Here, a new matrix is created with the same size and only one channel by initializing all the pixel values to 0. The obtained matrix with mean pixel values for the three channels is stored. All the pixel values are arranged as a 1D array by arranging one after another. Another feature that can be considered while differentiating the activity class can be a shape after color and size. The similar concept is to extract the object edges as features that can represent the change of another activity class. The edge of an activity class is identified by simply subtracting the values on either side of each pixel. The higher difference between the values can represent the edge with the significant transition at that pixel. Here, Prewitt kernel is applied by taking the values surrounding the selected pixel and multiplying with the kernel and adding the results to obtain the final value. Figure 2 shows sample images generated after applying various feature extraction techniques like shape, color, and edge detection.

4. CNN Model Design

Recurrent neural networks are different from traditional neural networks in that they use the output data from the previous step to the current step. They can be used for various tasks, such as emotion recognition. RNNs can also perform various tasks by working with sequences of data vectors. This feature allows them to remember a sequence

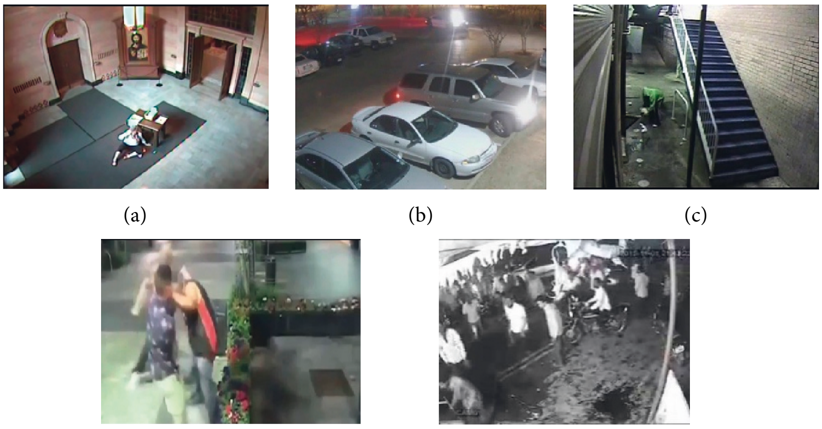


Figure 1: Sample images of five different human activities: (a) abuse, (b) arrest, (c) arson, (d) assault, and (e) fighting.

```
(i) Prerequisites: Generate image frames from input images and store them for further process.
(ii) Step 1. Iterate through all the images of a directory
(iii) Step 2. Generate the file hash using MD5 method for each image by dividing n-bit state into $n/4$ bit words and label it.
(iv) Step 3. Perform four stage n-bit message block process using nonlinear function $F$ and store it in a list.
(v) Step 4. Verify the image hash value in hash keys list.
(vi) Step 5. If exists, mark the image as duplicate.
(vii) Step 6. Remove the image from the dataset.
(viii) Step 7. Repeat the process from step 2 to step 6 for other dataset images
```

Algorithm 1: Removing duplicate image frames.

```
(i) Step 1. Select an image "img"
(ii) Step 2. Consider the dimension of img as $wx h$
(iii) Step 3.Generate number of features "nf" based on number of pixels using $nf = w \times h$
(iv) Step 4. Append pixel values one after each other
(v) Step 5. Generate feature vector based on nf
(vi) Step 6. Perform the feature extraction using f(img, dim)
```

Algorithm 2: Feature extraction using grayscale pixel values.

```
(i) Step 1. Load the RGB image with a dimension width $x$ height $x$ $c$ where "$c$" is the number of color channels
(ii) Step 2. Generate the number of features as a product of width, height, and $c$.
(iii) Step 3. Generate a new matrix by calculating the mean value of all the pixels
(iv) Step 4. Create a new image with same width and height by initializing all the values to 0.
(v) Step 5. Generate a 1D array by appending all the pixel values one after another.
```

Algorithm 3: Feature extraction using mean pixel values.

```
(i) Step 1. Select an image and identify the objects present using shape, color, and size features.
(ii) Step 2. Identify the edges by considering the values of each pixel. Higher variation in pixel values represent an edge. Conduct the
     edge detection using various kernels.
(iii) Step 2(a). Consider the values surrounded to each pixel and perform the multiplication with the kernel values.
```

Algorithm 4: Feature extraction using edge extraction.

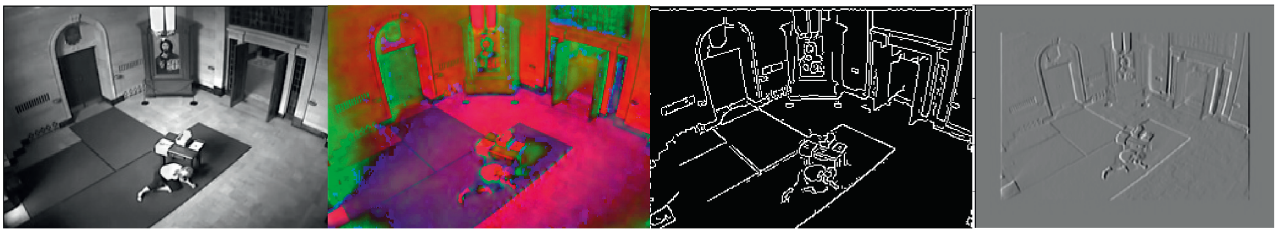


Figure 2: Sample image generated after applying feature extraction techniques.

of data. For instance, if a classification is performed on a set of data vectors, the output will depend on the previous results. RNN loses its effectiveness when the gap between the previous and the analyzed data increases. This is because while it can perform a task well for predicting sequences of data, it cannot remember the previous results. In the present study, researchers have developed a method that can solve this issue by developing a long-term memory architecture that can handle the missing dependencies.

A new DNN framework, called Deep Convolutional LSTM (DCLSTM), is introduced for human activity recognition and classification in this paper. Both convolutional and recurrent layers are combined in this design. The convolutional layers function as feature extractors and generate feature maps from the input image frames data. Features are activated in a time-dependent manner, and the recurrent layers model this process. The vector representation of LSTM network is shown as follows:

$$i_t = \sigma_i \left(w_{ai} a_t + w_{hi} h_{t-1} + w_{ci} c_{t-1} + b_i \right), \tag{1}$$

$$f_t = \sigma_f \left(w_{af} a_t + w_{hf} h_{t-1} + w_{cf} c_{t-1} + b_f \right), \tag{2}$$

$$c_t = f_t c_{t-1} + i_t \sigma_c \left(w_{ac} a_t + w_{hc} h_{t-1} + b_c \right), \tag{3}$$

$$o_t = \sigma_o \left(w_{ao} a_t + w_{ho} h_{t-1} + w_{co} c_{t-1} + b_o \right), \tag{4}$$

$$h_t = o_t \sigma_h \left(c_t \right), \tag{5}$$

where "i" is input, "o" is an output, "c" is cell activation, w is weight matrix, "b" is bias, and "h" is hidden value. The feature extraction using convolution neural network is given by

$$a_j^{(l+1)} (\tau) = \sigma \left(b_j^l + \sum_{f=1}^{F^l} K_{jf}^l (\tau) \times a_f^l (\tau) \right), \tag{6}$$

where $a_j^{(l+1)} (\tau)$ is feature map, σ is nonlinear function, "F" denotes feature maps, "K" is convolved kernel, and "P" is length of the kernel.

To conduct the classification process, two models named basic and proposed LSTM were considered. The basic LSTM model is a type of recurrent neural network that has 12 layers with 2-time distributed convolutional layers, 1 max pooling layer, 1-time distributed convolutional layer, 1-time distributed LSTM, 1 dropout layer, 1 LSTM layer, 1 dense layer, 1 LeakyReLU, 1 dropout, and 2 dense layers with 100 and 5 units, respectively.

The second LSTM model is the proposed model developed using 9 layers with 2-time distributed convolutional layer, 1 dropout layer, 1 max pooling, and 1 flatten layer. The last two layers are the dense layers with 100 and 5 units each. At the input layers, "ReLU" activation function was used, and at the classification layer, "softmax" classification function was used. The model is compiled with categorical cross-entropy loss value and an optimizer named "Adam" and extracted the "accuracy" performance metric. The model is evaluated using a batch size of 32 and an image size of 64 in width and 64 in height with 3 color channels. Table 1 depicts the summary of the proposed model and Figure 3 depicts the structure.

5. Experimental Outcomes and Discussions

This section describes the experiments in depth and presents the results of our analysis. To verify the proposed recognition method, we performed the experiments on the publicly available video dataset [UCF-Crime|Kaggle]. The dataset consists of 550 videos available for free download. The time duration of the videos is from 1–10 minutes and includes contents related to abuse, arrest, arson, assault, and normal scene videos. The videos are selected as input datasets to generate image frames and prepare training and testing datasets. The image frames are later picked randomly with equal number of images to balance the datasets. More than 18900 images were extracted from the input videos but only 2700 were considered for this study. Table 1 shows the details of datasets used during the experiments.

The original dataset includes new images generated after applying image augmentations like rescale, shift, shear, zoom, and flip. During the model design, several suitable hyperparameters like image size (64 width, 64 height), batch size (32), number of epochs (100), and filter size (3 × 3) are selected. The proposed model is a 12-layer architecture with three convolutional layers, three max pooling layers, three dropout layers, one LeakyReLU layer, and two dense layers. The model is evaluated up to 100 epochs due to the availability computational power. Table 2 shows the structure of the proposed model.

A random pickup process is followed to prepare the training and testing datasets by simply splitting the whole dataset into training (2250) and testing (450) with a split ratio of 80% and 20%. The entire dataset has five subdatasets for five different human activities with a 900 balanced set of images each. First, two experiments were conducted for each model using 100 epochs to have better results.

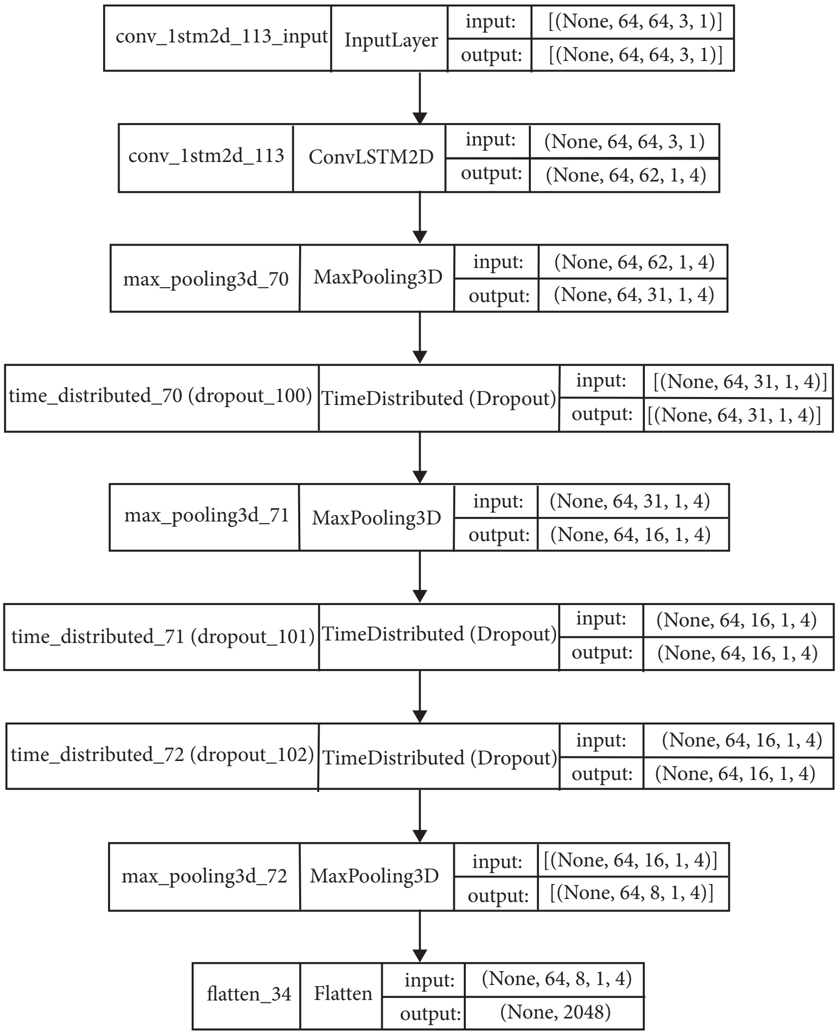


Figure 3: Structure of the proposed LSTM model.

Table 1: Training and testing datasets.

Image class	# of training images	# of testing images
Abuse	450	90
Arrest	450	90
Arson	450	90
Assault	450	90
Fighting	450	90
Total	2250	450

Table2: proposed LSTM model.

Layer (type)	Output shape	Param #
time_distributed_5 (TimeDistributed)	(None, none, 62, 64)	640
time_distributed_6 (TimeDistributed)	(None, none, 60, 64)	12352
time_distributed_7 (TimeDistributed)	(None, none, 60, 64)	0
time_distributed_8 (TimeDistributed)	(None, none, 30, 64)	0
time_distributed_9 (TimeDistributed)	(None, none, 1920)	0
lstm_1 (LSTM)	(None, 100)	808400
dropout_3 (dropout)	(None, 100)	0
dense_2 (dense)	(None, 100)	10100
dense_3 (dense)	(None, 5)	505

Table 3 presents the experimental outcomes obtained after running the activity recognition and classification process using basic LSTM and proposed LSTM models. Both models are trained and tested for 100 epochs to determine the model's efficiency.

When the basic model is implemented to perform the classification of five different human activities, its performance is very poor. Figure 4 shows the visualization of the performance of the basic and the proposed models. The training and the classification accuracies are just 18% and 21%, respectively. The training loss and classification loss are too high with unpredictable value and above 100%, respectively. The observations show that the basic LSTM model is neither efficient in training to human activities nor efficient in performing the classification.

On the other hand, when the proposed model is implemented to perform the classification of five different human activities, its performance is very good. The training

Table 3: Comparative analysis of classification accuracies.

S. No	LSTM models	# of epochs	Training accuracy (%)	Testing accuracy (%)	Training loss	Testing loss
1	Basic	100	18	21	Error	>100%
2	Proposed	100	99.68	100	NIL	0.016%

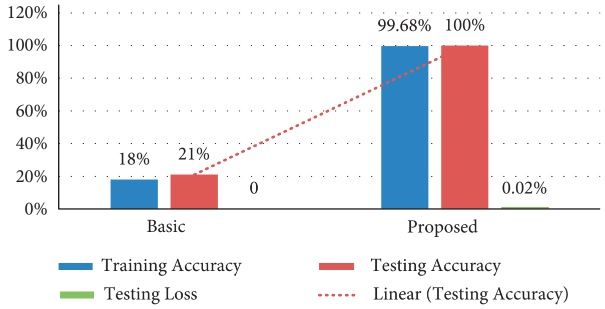
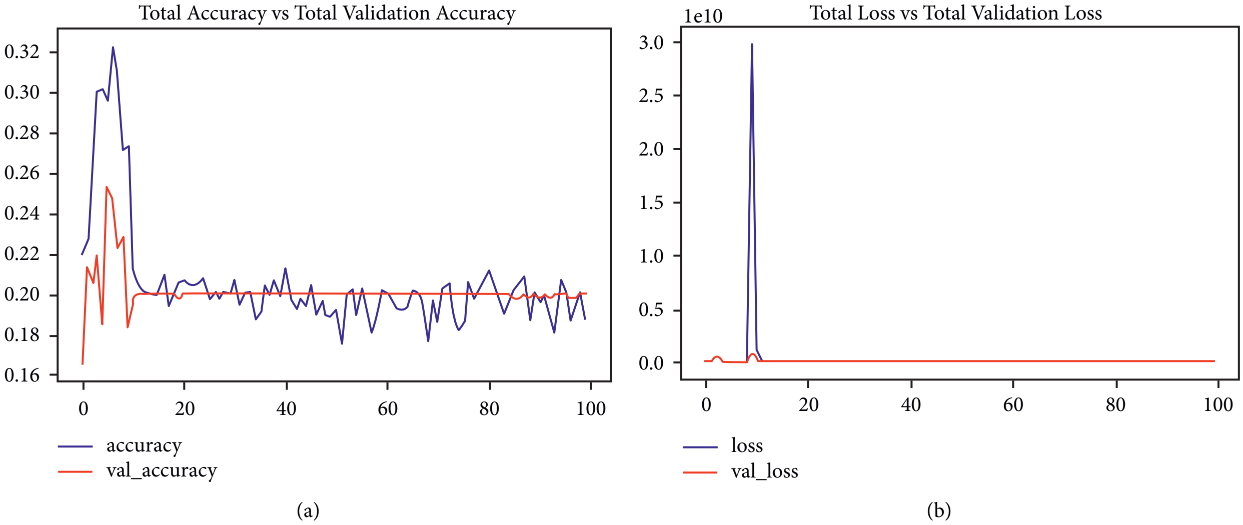


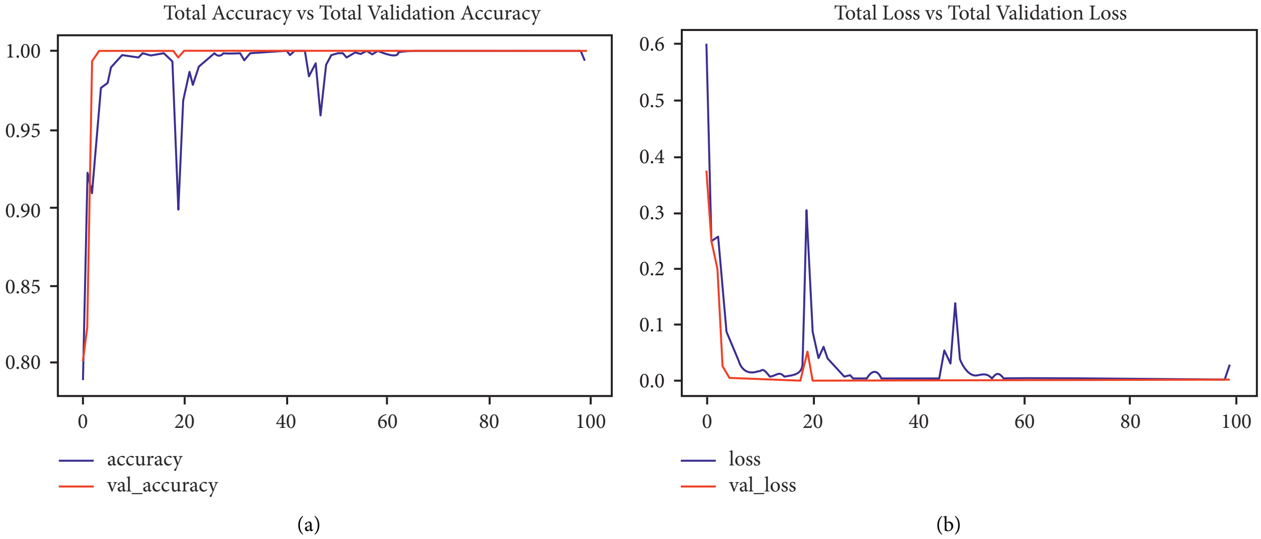
Figure 4: Performance comparison of the basic and the proposed model.



(a)

(b)

Figure 5: Performance measures of training and testing datasets: (a) accuracies and (b) loss.



(a)

(b)

Figure 6: Performance measures of training and testing datasets: (a) accuracies and (b) loss.

and testing accuracies are 99.68% and 100%, respectively. The training loss and classification loss are too low with no loss and 0.016%, respectively. The observations show that the proposed LSTM model is so efficient in training and understanding the human activities so that it can perform the classification well. The results mentioned in Table 2 are depicted in Figures 5 and 6.

6. Future Directions

The present study has opened a lot of challenges while working with human activity recognition application. The literature review has identified that a combination of CNN and LSTM while developing the neural network can give better results when compared with pure LSTM networks. But it has been observed that the five human activities considered in this study can only be possible and so efficient using pure LSTM model only. These observations show that further research to understand the behavior of models with CNN and LSTM can contribute best during recognition of human activities from the real-time videos. The further studies are planned to work more in developing new LSTM-based recurrent neural networks models to perform the recognizing human activities even from huge size of videos. The study is also looking into the other performance metrics like precision, recall, and F1-score values that can also guide the performance of any LSTM model. Later, a comparative analysis will be considered to determine the efficiency of the newly developed model by comparing the results with other existing models.

7. Conclusion

Deep learning has recently been used to overcome difficulties in image processing, natural language processing, and speech recognition. Human activity recognition (HAR) is a hot research area among scientists and engineers. Deep learning can now address HAR difficulties, thanks to two new study areas. Recurrent neural networks (RNNs) are better at detecting anomalous human behavior and preventing security breaches. This article offers a deep network design using residual bidirectional long-term memory (LSTM). The novel network can prevent gradient vanishing in time and space to increase recognition rates. The basic and proposed LSTM models were utilized to investigate activity recognition and classification. The algorithms' efficacies in classifying five human acts such as abuse, arson, assault, and fighting images are then compared. With larger training and classification loss levels, the simple LSTM model improved training accuracy to 18% and testing accuracy to 21%. The proposed LSTM model outperformed the simple model, achieving 100% classification accuracy. Finally, the suggested LSTM model performs well in real-time video recognition and classification.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition inUbiquitous computing," in *Proceedings of the Twenty-Second International Joint Conferenceon Artificial Intelligence*, pp. 1729–1734, Barcelona, Catalonia, Spain, July 2011.

[2] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep ActivityRecognition models with triaxial accelerometers," in *Proceedings of the AAAI Workshop: 13Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, Phoenix,AZ, USA, February 2016.

[3] M. Zeng, L. T. Nguyen, B. Yu et al., "ConvolutionalNeural networks for human activity recognition using mobile sensors," in *Proceedings Ofthe 6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205, Austin,TX, USA, November 2014.

[4] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based onSingle accelerometer," in *Proceedings of the IEEE International Conference on Systems,Man, and Cybernetics*, pp. 1488–1492, Hong Kong, China, June 2015.

[5] H.-O. Hessen and A. J. Tessem, *Human Activity Recognition with Two Body-WornAccelerometer Sensors*, Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2015.

[6] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neuralnetworks on multichannel time series for human activity recognition," in *Proceedings of The24th International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, July 2015.

[7] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: aresource efficient implementation on low-power devices," in *Proceedings of the IEEE 13thInternational Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 71–76, SanFrancisco, CA, USA, June 2016.

[8] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition usingtime-delay neural networks," *IEEE Transactions on Acoustics, Speech, & Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[9] O. Ramana Murthy and R. Goecke, "Ordered trajectories for human action recognition with large number of classes," *Image and Vision Computing*, vol. 42, pp. 22–34, Oct. 2015.

[10] U. Amin, J. Ahmad, M. Khan, M. Sajjad, and W. B. Sung, "Action recognition in video sequences using deep Bi-directional LSTM with CNN features, special section on visual surveillance and biometrics: practices, challenges, and possibilities," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[11] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, https://arxiv.org/abs/1506.00019.

[12] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *in Proceedings of the Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, 15th International Conference*, p. 753, Warsaw, Poland, September 2005.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.