Journal of Engineering
and Applied Science

RESEARCH Open Access

A novel human activity recognition architecture: using residual inception ConvLSTM layer

Sarah Khater*, Mayada Hadhoud and Magda B. Fayek

*Correspondence: srashad@cu.edu.eg
Computer Engineering Department, Faculty of Engineering, Cairo University, Cairo, Egypt

Abstract

Human activity recognition (HAR) is a very challenging problem that requires identifying an activity performed by a single individual or a group of people observed from spatiotemporal data. Many computer vision applications require a solution to HAR. To name a few, surveillance systems, medical and health care monitoring applications, and smart home assistant devices. The rapid development of machine learning leads to a great advance in HAR solutions. One of these solutions is using ConvLSTM architecture. ConvLSTM architectures have recently been used in many spatiotemporal computer vision applications.

In this paper, we introduce a new layer, residual inception convolutional recurrent layer, ResIncConvLSTM, a variation of ConvLSTM layer. Also, a novel architecture to solve HAR using the introduced layer is proposed. Our proposed architecture resulted in an accuracy improvement by 7% from ConvLSTM baseline architecture. The comparisons are held in terms of classification accuracy. The architectures are trained using KTH dataset and tested against both KTH and Weizmann datasets. The architectures are also trained and tested against a subset of UCF Sports Action dataset. Also, experimental results show the effectiveness of our proposed architecture compared to other state-of-the-art architectures.

Keywords: HAR, Residual, Inception, ConvLSTM, KTH

Introduction

Machine learning overlaps with many science fields. One of these fields is computer vision [1]. Machine learning methods provide computer vision with techniques that offer a great leap in many of its applications, for instance, object recognition, tracking, classification, 3D modeling, and many other applications. These applications have witnessed great improvement with the emergence of new machine learning techniques. One of these applications is HAR. HAR is the basis for many other computer vision applications, for example augmented reality, robotics, automatic monitoring for medical, industrial, or surveillance purposes, etc. [2]. HAR aims at recognizing an action from data acquired by sensors, images, videos, etc. Proposed HAR systems face many limitations that narrow their applicability on domain-specific applications: limitations like (i) lighting variation, (ii) camera perspective change, (iii) scale variation, (iv) occlusion resulting from same human body parts or even from surrounding objects, (v) background confusion by undefined static or moving objects, and (vi) class confusion whether from intra-class similarities or inter-class variation [3]. Also another limitation that may rise with training of domain-specific applications is the lack of descriptive datasets. Also, proposed HAR systems face many challenges such as the following: (i) Reliability: HAR systems are required to make continuous real-time decisions based on what the system yields in surveillance applications. (ii) Social acceptance: as in smart home assistant devices, these devices closely monitor users behaviors at home, so this invades the users privacy, which raises the disclosure challenge. (iii) Disclosure: the data is required to be processed on the same device and not to be shared with any other parties for processing or advertisement. Due to the multidisciplinary nature of HAR [2], HAR requires continuous examination and exploration of new solutions to tackle these limitations and challenges.

To overcome the limitations and challenges of solving HAR problems, new techniques must be investigated. One of the most successful and recent technique is ConvLSTM [4]; ConvLSTM is a convolution LSTM layer used in many computer vision applications that require processing of spatiotemporal data. ConvLSTM uses convolution operation on an LSTM unit to perform input-to-state transformation or state-to-state transformation. ConvLSTM accepts a set of data over time to perform some prediction task. ConvLSTM has been widely used in many computer vision applications, for example, object tracking [5], scene segmentation [6], activity recognition [7], and video enhancement, like video rain removal [8]. In this paper, we propose a modification to conventional ConvLSTM layer, inspired by [9]. The novelty of our approach lies in the following points:

1 The introduction of residual inception convolutional recurrent layer, ResIncConvLSTM, a variation of conventional ConvLSTM layer that incorporates both the concepts of residual and inception with ConvLSTM layer
2 Solving Human activity recognition problem by designing an architecture using the newly introduced ResIncConvLSTM layer

Our proposed approach is found to outperform state of the art architecture [4] by 7% when tested on KTH dataset [10] and 11% when tested on Weizmann dataset [11]. We trained and tested our approach on a subset of UCF Sports Action dataset [12, 13] and it is found to outperform state of the art architecture [4] by 21%. Also, our proposed architecture shows efficiency improvement compared to some state-of-the-art architectures. The fact that the effectiveness of our proposed architecture is better than both the ConvLSTM baseline architecture and some state-of-the-art architectures promises better results by replacing some of the conventional ConvLSTM layers with the proposed ResIncConvLSTM layer in deeper ConvLSTM-based architectures.

Related work

HAR is a core component in various computer vision applications, so an extensive amount of research is done to tackle HAR problem. In this section, we briefly review the most remarkable and related work done to solve HAR problem. And since in this paper, we only examine visual data, we are only concerned with vision-based HAR methods. HAR approaches can be classified according to the following perspectives.

Feature extraction process

One perspective to classify HAR approaches is the process by which the features are extracted. The process of extracting features is either manually crafting features, or by using classical machine learning for automatically learning these features. Manually crafting features can either use the external seen features of the object just like body parts or motion [14–16] or a hybrid between these features [17, 18]. Many feature extraction-based methods are recently used to solve HAR. Nazir et al. [19] proposed a novel feature representation, 3D Harris space-time interest point detector and 3D Scale-Invariant Feature Transform (3DSIFT) descriptor. The features are then extracted and arranged for each frame sequence using Bag-of-features (BOF) method. Then a multiclass support vector machine (SVM) model is trained and tested using the extracted BOF. Nadeem et al. [20] proposed a HAR method based on training an artificial neural network with multidimensional features. These multidimensional features are estimated for twelve different body parts using body models. On the other hand, automatic learning of features can be done using a non-deep learning approaches like Bayesian network [21] or dictionary learning [22], or using deep learning. CNN [23], RNN [24], ConvLSTM, and CNN-RNN [25, 26] are examples of deep learning approaches. Since the introduction of ConvLSTM layer, many ConvLSTM-based architectures are introduced to solve HAR problem. In 2018, Yuki et al. [27] proposed dual-ConvLSTM to extract global and local features and use them in the recognition process. In 2019, Majd and Safabakhsh [7] introduced motion-aware ConvLSTM architecture that captures not only spatial and temporal correlations but also motion correlations between successive frames. In 2020, Kwon et al. [28] introduced hierarchical deep ConvLSTM architecture to capture different complex features.

Feature representation

One perspective to classify HAR approaches is how features are represented. Features can be arranged to preserve spatial, temporal, color, or dimensional information. The features can be flattened as in BOF method losing its spatial information. Aly and Sayed [29] proposed a HAR method that exploits both global and local features to differentiate between similar actions, like walking and running. The method first extracts from the input sequence of frames both global and local Zernike Moment features. Theses features are then combined and represented using BOF method. Then multiclass SVM algorithm is trained to recognize different actions. Another example of feature representation is silhouette frames. The input undergo silhouette extraction. Silhouette image or frame is an image that shows only the outline or the shape of the object of interest. The most important characteristic in silhouette representation is recognizing moving objects. Ramya and Rajeswari proposed, in [30], a HAR method based on silhouette frames. The method consists of three consecutive stages. First, the method preforms background subtraction to extract silhouette frames. Second, distance transform based features and entropy features are extracted from the silhouette frames. Third, a neural network is trained using these extracted features to recognize various actions. Another feature representation is bit maps, for example, features can be arranged to be a bit map of the moving object of interest. In [31], the authors proposed a solution to HAR problem using 3D CNN network receiving as an input a 3D motion cuboid. Input binarization is also one type of feature representation. Another famous feature representation is training the model with a sequence of sampled frames, where each pixel in each frame is considered a feature value. [23, 24, 32, 33] are examples for approaches that use a sequence of sampled frames as the input features.

Supervision Level

Another perspective to classify HAR approaches is whether the approach is supervised, unsupervised, or semi-supervised learning. Supervised learning approaches relies on training the model using massive well-labeled data. [34, 35] are examples of HAR supervised approaches. In [34], Han et al. proposed a two stream CNN model to perform action recognition. The authors proposed an augmentation strategy to overcome the overfitting problem that arises from the absence of massive labeled datasets. The proposed augmentation strategy is based on remodeling the dataset using a transfer learning model. In [35], Zhang et al. proposed a feature extraction approach that decouples spatial and temporal features. The proposed approach is based on a dual channel feedforward network to extract static spatial features from a single frame and dynamic temporal features from consecutive frame differences. Both decoupled spatial and temporal features are then fed to a multiclass SVM model for action recognition. On the other hand, unsupervised learning do not need labeled data. Discrimintive features are learned by the model from unlabeled data. [36–42] are examples of work done to solve HAR problem based on unsupervised learning. Abdelbaky and Aly [36–39] presented several solutions to HAR based on unsupervised deep convolution network PCANet, a simple PCANet [36], and an extended solution in which spatiotemporal features are learned from three orthogonal planes (TOP), PCANet-TOP [38]. In [40, 41], Rodriguez et al. solved HAR problem using one-shot learning in which a class representation is built from a few or a single training sequence. The authors proposed a model based on Simplex Hidden Markov Model (SHMM) and an optimized Fast Simplex Hidden Markov Model (Fast-SHMM). In [42], Haddad et al. presented a method to solve HAR problem using one-shot learning model using Gunner Farneback's dense optical flow (GF-OF), Gaussian mixture models, and information divergence. Semi-supervised learning is a hybrid approach that benefits from the ability of supervised learning to learn features and the ability of unsupervised learning to learn hidden non-visual patterns. In semi-supervised learning, the training is done with partially labeled data. Also, not all the classes are essentially known. [43, 44] are examples of work done to solve HAR problem based on semi-supervised learning approach.

Essential background

In this paper, a new layer is proposed by integrating residual and inception concepts into ConvLSTM layer. In this section, a brief introduction about ConvLSTM, residual, and inception architectures are explained.

ConvLSTM architecture

ConvLSTM is first introduced in [4]. ConvLSTM overcomes the shortcoming of fully connected LSTM in handling spatial data. Fully connected LSTM handles temporal correlation leaving out encoding spatial data. ConvLSTM addresses this problem by applying

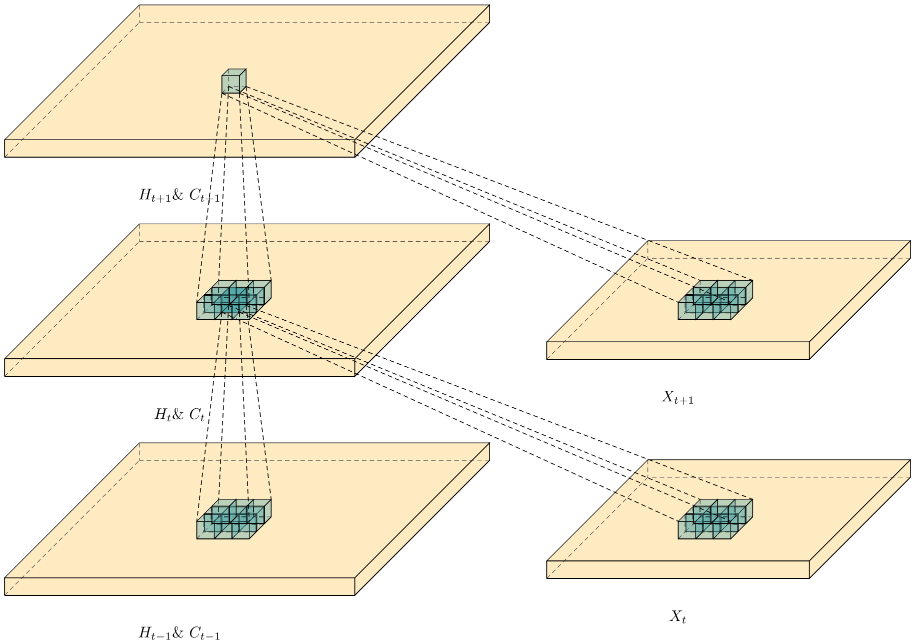

Fig. 1 ConvLSTM unfolded structure

convolution to input-to-state and state-to-state transformations. Figure 1 illustrates ConvLSTM unfolded structure and its main tensors, the inputs $X_1, ..., X_t$, cell current states C_1, ...,C_t, and hidden states H_1,..., H_t, which are also considered cell output.

(1)–(5) are the fundamental equations of ConvLSTM. Gates i_t, f_t, o_t are 3D tensors. "σ" is nonlinear activation function, "$*$" is convolution operator, and "$\circ$" is the Hadamard product. The transformation from one state to another occurs as illustrated in (1)–(5). The next value of each cell in the grid is determined by both the input and the current value of the neighboring cells. This can be achieved by applying convolution to input-to-state and state-to-state transformations.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{2}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \tag{4}$$

$$H_t = o_t \circ tanh(C_t) \tag{5}$$

Padding is used to make sure that both hidden states and inputs have the same dimensions. Padding of hidden states on the borders is viewed as using the state of the outside world in calculation. Hidden states are initialized with zeros to indicate total ambiguity of the future.

ConvLSTM-based architecture is first applied on next frame prediction using artificially generated moving MNIST [45] dataset in [4]. Since the introduction of ConvLSTM layer in 2015, many modifications were proposed to improve the layer architecture. The same exact year, Liang and Hu [46] introduced simple recurrent convolutional unit, RCNN, and applied it on objet recognition using SVHN [47], CIFAR-100 [48], and MNIST [49] datasets. In 2017, Alom et al. [50] proposed inception recurrent convolutional unit, IRCNN. In 2018, Wei et al. [51] introduced residual ConvLSTM and applied it on tweet count prediction. In 2020, Alom et al. [9] proposed residual inception recurrent unit, IRRCNN. IRCNN and IRRCNN are tested against CIFAR-100.

Inception architecture

Inception architecture is first introduced in [52]. Inception module is designed as shown in Fig. 2. It combines output from 3 different layers (1 × 1 conv, 3 × 3 conv, and 5 × 5 conv) and concatenates them to form the input to the next layer. Different filter sizes help detecting features that may come in different sizes. The 1 × 1 conv that precedes 3 × 3 conv and 5 × 5 is used to compute reductions before the computationally expensive layers 3 × 3 conv and 5 × 5 conv. Inception modules are recommended to be used in higher layers to extract complex features, while using conventional convolutional layers as lower layers.

Residual architecture

Residual architectures are introduced in [53]. Theoretically, as a neural network goes deeper, a more complex nonlinear function is learned. This complex function becomes

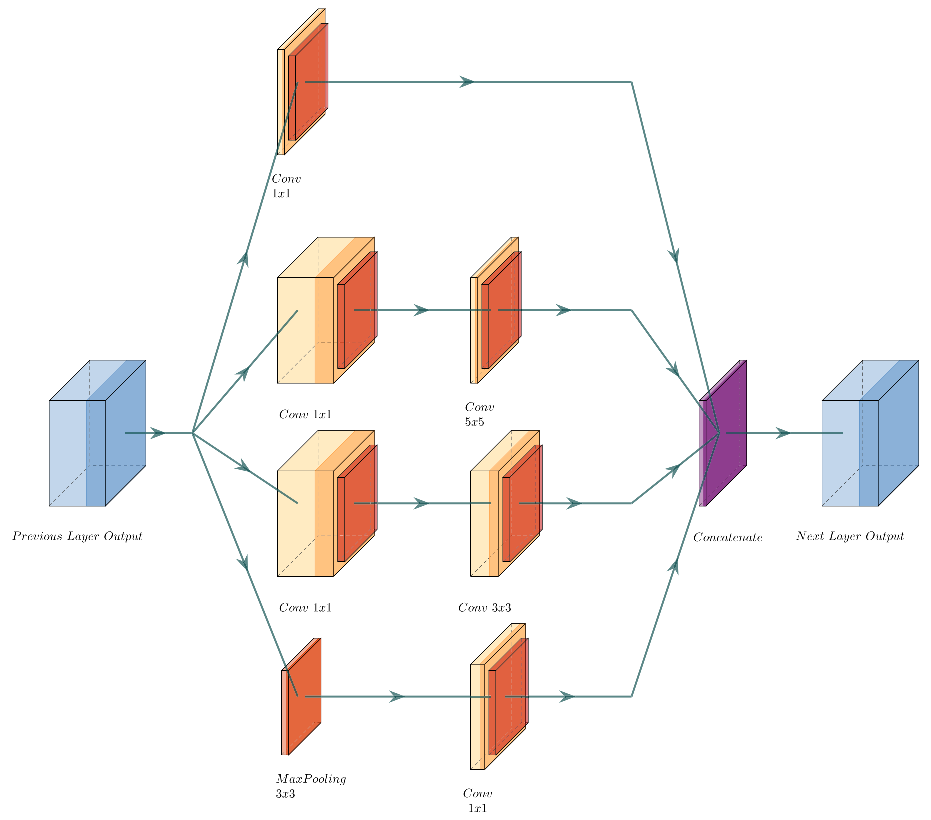

Fig. 2 Inception architecture

able to discriminate between different classes easily. For applying this practically, sufficient dataset should be used in training this network. Training takes place by updating the current weights with values proportional to the gradient of the loss function. Also, as a neural network goes deeper a problem with gradient flow arises. The gradient becomes vanishingly small. This prevents the neural network from any further change to the network weights, so eventually the network does not learn anything new. This problem is called the vanishing gradient problem. Residual architectures are motivated by this problem. A residual neural network is based on the idea of shortcuts to skip some layers, typically two or three layers, as shown in Fig. 3. This helps with the vanishing gradient problem by reusing activation from previous layers until the adjacent layers learn and adjust their weights. This allows an alternative way for the gradient to flow.

Methods

In this section, the proposed ResIncConvLSTM layer is explained in more details. The proposed layer incorporates residual and inception concepts into the conventional ConvLSTM.

ResIncConvLSTM layer design

Figure 4 shows the design of ResIncConvLSTM layer. The input from the previous layer, X, is fed to the ResIncConvLSTM layer. ResIncConvLSTM consists of two main parts. The first part is the inception part. The inception part consists of four parallel branches:

1 A ConvLSTM layer with kernel size 1 × 1 and number of filters f_{a1x1} , $H_a(X)$

2 A ConvLSTM layer with kernel size 1 × 1 and number of filters f_{b1x1}, followed by a ConvLSTM layer with kernel size 3 × 3 and number of filters f_{b3x3}, $H_b(X)$

3 A ConvLSTM layer with kernel size 1 × 1 and number of filters f_{c1x1}, followed by a ConvLSTM layer with kernel size 5 × 5 and number of filters f_{c5x5}, , $H_c(X)$

4 A Maxpooling layer with kernel size 3 × 3, stride of 1 and "same" padding, followed by a ConvLSTM layer with kernel size 1 × 1 and number of filters f_{d1x1}, $H_d(X)$

These four branches produce intermediate outputs $H_a(X)$, $H_b(X)$, $H_c(X)$, and $H_d(X)$, considering that all the branches receive the same input, X.

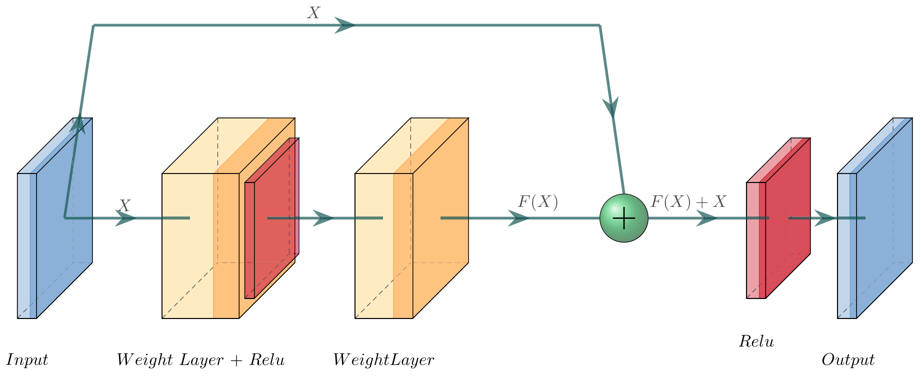

Fig. 3 Residual architecture

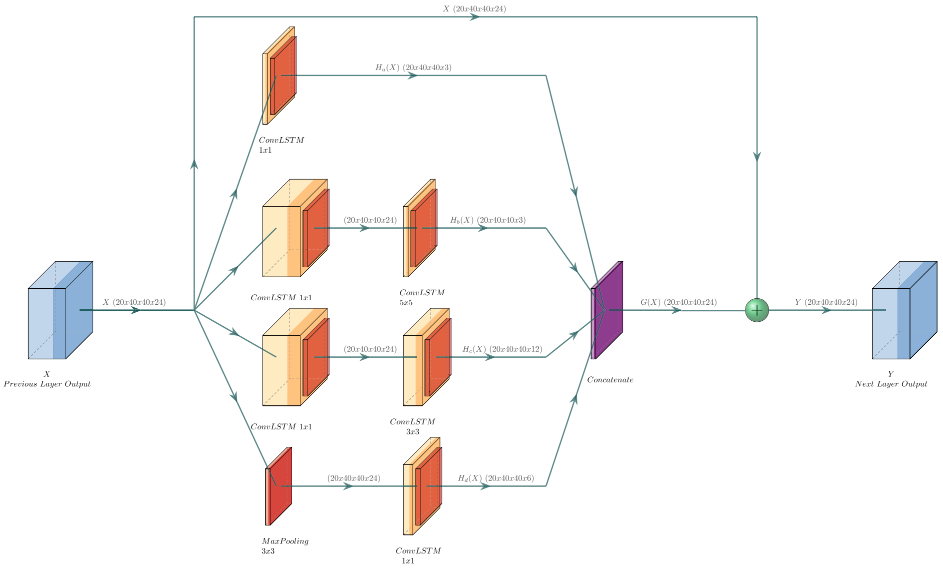

Fig. 4 Proposed ResIncConvLSTM structure labeled with data tensor dimensions and corresponding equations notation

These intermediate outputs are then concatenated to form the output to this part, $G(X)$. The second part, residual part, adds both the original input, X, with the output from inception part, $G(X)$, to produce the final output, Y.

The design in Fig. 4 can be described using (6) and (7). "$\bigodot$" is concatenation operator. "+" is addition operator.

$$G(X) = H_a(X) \bigodot H_b(X) \bigodot H_c(X) \bigodot H_d(X) \tag{6}$$

$$Y = G(X) + X \tag{7}$$

Solving HAR using ResIncConvLSTM layer

Figure 5 shows the proposed architecture. The architecture uses ResIncConvLSTM layer as its fundamental module. The architecture is arranged as follows:

1 A conventional ConvLSTM layer was used as the lower layer to capture low level features, it also transforms the input shape tensor dimensions from (20, 40, 40, 1) to (20, 40, 40, 24), this dimension expansion improves the process of extracting and learning of new features in subsequent layers, and this expansion cannot be done

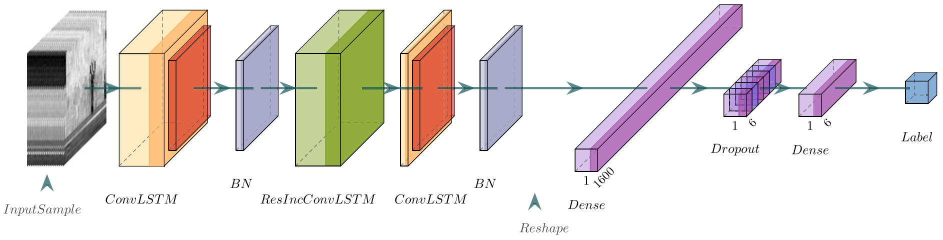

Fig. 5 Proposed model using ResIncConvLSTM layer

using ResIncConvLSTM layer as in residual layers the input and the output tensor dimensions should conform,

2 Followed by ResIncConvLSTM, to capture higher level features, then

3 Another conventional ConvLSTM layer, this layer reduces the dimensions from (20, 40, 40, 24) to (40, 40, 1); in other words, it returns a single reduced tensor for every input sequence, and finally

4 A stack of dense fully connected layers with drop out.

This design is trained against ConvLSTM baseline architecture which is a stack of conventional ConvLSTM layers followed by a stack of dense fully connected layers with dropout proposed in [4]. ConvLSTM baseline architecture is adequate in comparisons because it is simple to make the evaluation of our proposed work unbiased and not affected by any assisting factors.

Both models are trained for 20 epochs using the same dataset. Our model has 52,108 trainable parameters. ConvLSTM baseline model has 70,492 trainable parameters. Tables 1 and 2 show the detailed description of the proposed architecture and ConvLSTM baseline architecture, respectively. The tables describe each layer with its input, input tensor size, and output tensor size. The tables show also whether a ConvLSTM layer returns the whole sequence or not.

Data description

The input is arranged in a 5D tensor form: (B, T, W, H, ch), where B is the batch size, T is the number of frames per sequence, in other words the number of timesteps, W is the width of the frame, H is the height of the frame, and ch is the number of channels. The output is either a 5D tensor with dimensions (B, T, W, H, f) or a 4D tensor with dimensions (B, W, H, f), where f is the number of filters and B, T, W, and H as illustrated before. The 5D tensor case is when there is an output for each timestep. The 4D tensor case is when a single output is returned for the whole sequence. The dimensions of the output depends on the nature of the application. In case of HAR, only one final output, single label, is required for the whole sequence, so the output is either a 4D tensor, or a

Table 1 Proposed model detailed description

Layer ID	Layer name	Input size	Output size	Number of parameters	Input ID	Return sequence
system_input	Input	(16, 20, 40, 40, 1)	(16, 20, 40, 40, 1)	0	-	-
convlstm_m_1	ConvLSTM	(16, 20, 40, 40, 1)	(16, 20, 40, 40, 24)	21,696	system_input	Yes
batch_norm_1	Batch Normalization	(16, 20, 40, 40, 24)	(16, 20, 40, 40, 24)	96	convlstm_m_1	-
resincconvlstm	ResIncConvLSTM	(16, 20, 40, 40, 24)	(16, 20, 40, 40, 24)	19,760	batch_norm_1	No
convlstm_m_2	ConvLSTM	(16, 20, 40, 40, 24)	(16, 40, 40, 1)	904	resincconvlstm	No
batch_norm_2	Batch Normalization	(16, 40, 40, 1)	(16, 40, 40, 1)	4	convlstm_m_2	-
reshape	Reshape	(16, 40, 40, 1)	(16, 1, 1600)	0	batch_norm_2	-
dense_1	Dense	(16, 1, 1600)	(16, 1, 6)	9606	reshape	-
dropout	Dropout	(16, 1, 6)	(16, 1, 6)	0	dense_1	-
dense_1	Dense	(16, 1, 6)	(16, 1, 6)	42	dropout	-

Table 2 ConvLSTM baseline architecture detailed description

Layer ID	Layer name	Input size	Output size	Number of parameters	Input ID	Return sequence
system_input	Input	(16, 20, 40, 40, 1)	(16, 20, 40, 40, 1)	0	-	-
convlstm_1_3x3	ConvLSTM	(16, 20, 40, 40, 1)	(16, 20, 40, 40, 40)	59,200	system_input	Yes
batch_norm_1	Batch Normalization	(16, 20, 40, 40, 40)	(16, 20, 40, 40, 40)	160	convlstm_1_3x3	-
convlstm_2_3x3	ConvLSTM	(16, 20, 40, 40, 40)	(16, 40, 40, 1)	1480	batch_norm_1	No
batch_norm_2	Batch Normalization	(16, 40, 40, 1)	(16, 40, 40, 1)	4	convlstm_2_3x3	-
reshape	Reshape	(16, 40, 40, 1)	(16, 1, 1600)	0	batch_norm_2	-
dense_1	Dense	(16, 1, 1600)	(16, 1, 6)	9606	reshape	-
dropout	Dropout	(16, 1, 6)	(16, 1, 6)	0	dense_1	-
dense_1	Dense	(16, 1, 6)	(16, 1, 6)	42	dropout	-

5D tensor reduced to a 4D tensor in a subsequent layer. In our proposed model, we chose the latter choice. In other applications, like next frame prediction or object tracking, the output is a 5D tensor because an output is required for each time step.

Results and discussion

This section discusses the setup and the results of the experiments held to show the significance of the proposed work. The experiments are run on a computer with quad-core Intel Core i7 processor running at 2.3 GHz using RAM of 8GB of 1600MHz DDR3 memory.

Datasets

The benchmark used to train our model is KTH dataset [10]. The KTH dataset is a human action video database of six human actions (walking, jogging, running, boxing, two hands waving, and hand clapping). We used KTH in the training and evaluation process because it is one of the biggest human activity dataset. The dataset contains 2391 sequences of around 4 s on average, filmed with a static camera over a homogeneous background, indoor and outdoor settings. The model is tested against both KTH and Weizmann [11] datasets. Weizmann is a human action dataset of 90 sequences of nine actors with ten human actions (walking, running, jumping, galloping sideways, bending, one-hand waving, two-hands waving, jumping in place, jumping jack, and skipping). Only actions similar to the KTH dataset are considered in the evaluation process (walking, running, and two-hands waving). Also, 10% of the KTH dataset is set aside and never used during training to be used in the evaluation process. Also, we trained and tested our approach against a subset of UCF Sports Action dataset. UCF Sports Action dataset is a public dataset collecting a set of 10 actions from different sports that are originally broadcasted on television. The actions included in the dataset are diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swing-bench, swing-side, and walking. We only considered the following actions in the evaluation process: diving, lifting, riding horse, swing-side, and walking. The dataset contains 150 sequence of duration from around 2 to 14 s. Table 3 concludes KTH, Weizmann, and UCF Sports Action datasets description. The training data are sampled at 20 frames per video, resized to grayscale 40 × 40 pixels. The training data is preprocessed, resized, and converted to grayscale frames, to reduce

Table 3 KTH, Weizmann, and UCF Sports Action datasets' description

	Number of actions	Number of videos	Average video length
KTH	6	2391	4 s
Weizmann	10	90	1.5 s
UCF Sports Action	10	150	2.20 to 14.40 s

the number of trainable parameters due to our hardware computational limitations. Data augmentation methods are used to expand the dataset. The following data augmentation methods are used: addition of Gaussian noise, flip, shifting, zooming, and rotation by angles between 2 and 12°. Our model is trained and tested on single individual action detection. The model accepts a sequence of frames and outputs a single output for the whole sequence labeling the human activity recognized from the sequence.

Experiments

This section discusses the results of the experiments held on both ResIncConvLSTM-based and ConvLSTM baseline models. Both models are trained for 20 epochs using KTH dataset. Figure 6 shows training and validation losses of our proposed architecture. The figure shows that loss in both training and validation decrease gradually in a similar manner with the successive epochs. This shows that the architecture does not overfit to the training data and actually does learn distinctive features.

Validation accuracy can be shown in Fig. 7. The figure shows how accuracy of ResIncConvLSTM architecture is lower than ConvLSTM baseline model in the early epochs and then ResIncConvLSTM architecture gradually exceeds ConvLSTM baseline architecture.

Tables 4 and 5 show the confusion matrices of both ResIncConvLSTM and ConvLSTM baseline architectures on KTH dataset. The tables show that both architectures perform well in recognizing boxing, clapping, and waving activities with accuracies 99.9%, 99.5%, and 99.7%, respectively, for ResIncConvLSTM architecture, and 98.1%, 98.4%, and 98.9%, respectively, for ConvLSTM baseline architecture. ResIncConvLSTM performs

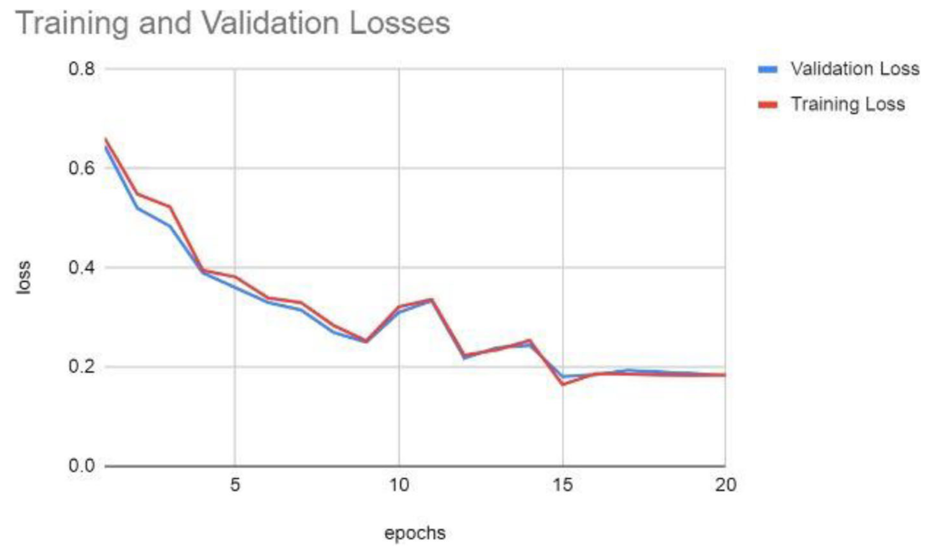

Fig. 6 Training and validation losses

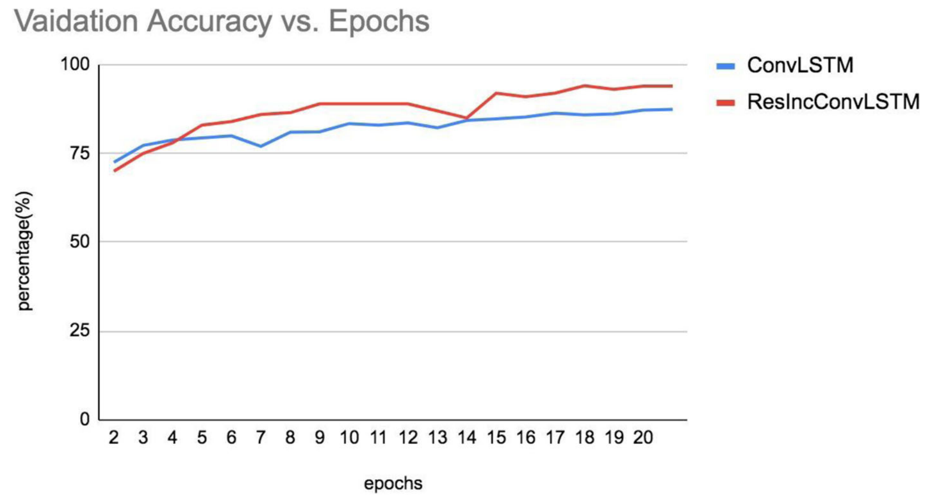

Fig. 7 Validation accuracy vs. epochs

slightly better in recognizing boxing, clapping, and waving. Performing well for both architectures is quite expected because each of these activities has distinctive features. ResIncConvLSTM performs better by 10.4%, 14.7%, and 12% in recognizing jogging, running, and walking activities, respectively, with accuracies 79.9%, 82.9%, and 98.1% versus 69.5%, 68.2%, and 86.1%, respectively. The confusion matrix shows how ResIncConvLSTM performance decreases, compared to boxing, clapping, and waving, with activities jogging and running. Both activities are confused with each other and with walking because those activities have similar features. Overall, ResIncConvLSTM performs better than ConvLSTM baseline architecture by 7% with classification accuracies 94.08% versus 87% when tested against KTH dataset. ResIncConvLSTM performs better than ConvLSTM baseline architecture by 11% with classification accuracies 79% versus 68% when tested against Weizmann dataset for the common action classes, walking, running, and two-hands waving. Also, ResIncConvLSTM-based architecture is found to outperform ConvLSTM baseline architecture by 21% when tested against a subset of UCF Sports Action dataset with accuracies 68.8% versus 47.66%. The experiment objective is to prove that the proposed layer, ResIncConvLSTM, still outperforms conventional ConvLSTM. Although, our ResIncConvLSTM-based architecture performed better than ConvLSTM-based architecture in classifying the selected actions, this experiment provide us with the following insight for future research: bigger datasets require deeper networks and bigger input size.

Table 4 ResIncConvLSTM architecture confusion matrix on KTH dataset

	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	0.999	0.004	0.001	0	0	0
Clapping	0	0.995	0.002	0	0	0
Waving	0.001	0.001	0.997	0	0	0
Jogging	0	0.001	0	0.799	0.17	0.019
Running	0	0	0	0.083	0.829	0
Walking	0	0	0	0.118	0.001	0.981

Table 5 ConvLSTM baseline architecture confusion matrix on KTH dataset

	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	0.981	0.001	0.007	0	0.004	0
Clapping	0.006	0.984	0.004	0	0.026	0
Waving	0.012	0.014	0.989	0	0	0
Jogging	0	0	0	0.695	0.204	0.085
Running	0.001	0.001	0	0.171	0.682	0.054
Walking	0	0	0	0.134	0.084	0.861

Comparison with existing approaches

Table 6 shows a comparison between different state-of-the-art approaches used to solve HAR problem, including our proposed approach, evaluated on KTH dataset. The comparison is held in terms of classification accuracy. Based on the presented comparison, our proposed approach outperforms [19, 20, 29, 30, 34–39, 41, 42] and shows a comparable classification accuracy with [31, 32]. Although classification accuracy of our proposed approach is less than [31, 32], by less than 1%, our proposed approach yields good results using a relatively small number of parameters.

Conclusions

In this paper, a novel layer based on residual and inception is introduced, ResIncConvLSTM layer. The proposed layer is used for solving HAR problem. A ResIncConvLSTM-based architeture is designed and trained on KTH dataset. The designed architecture is tested on KTH, Weizmann, and UCF Sports Action datasets. ResIncConvLSTM-based architecture is found to perform better than ConvLSTM baseline architecture by 7%, 11%, and 21% on KTH, Weizmann, and UCF Sports Action datasets, respectively. Also, experimental results show the effectiveness of our proposed architecture compared to other state-of-the-art architectures. Our future work will concentrate on applying ResIncConvLSTM on different computer vision applications like segmentation, next frame prediction, object tracking, and scene summarization. We may also investigate dual model approach to solve multiple individual HAR with ResIncConvLSTM.

Table 6 Comparison with existing approaches on KTH dataset

Reference	Method	Publication year	Accuracy(%)
Haddad et al. [42]	GF-OF and GMM	2021	73.1%
Abdelbaky and Aly [36–39]	PCANet	2020-2021	85.5%-93.3%
Ramya and Rajeswar [30]	Distance Transform + Entropy Features + ANN	2021	91.4%
Nadeem et al. [20]	SVM + ANN	2020	87.57%
Aly and Sayed [29]	Zernike Moment + BOF + SVM	2019	81.03%
Han [34]	Two-stream CNN	2018	93.1%
Nazir et al. [19]	3DHarris + 3DSIFT + BOF + SVM	2018	91.82%
Zhang et al. [35]	Dual-channel Deep Network	2018	92.8%
Rodriguez et al. [41]	Fast-SHMM	2017	74%
Abdekkaoui and Douik [32]	DBN	2020	94.83%
Arunnehru et al. [31]	3D CNN + 3D motion cuboid	2018	94.9%
Proposed approach	ResIncConvLSTM	2021	94.08%

Abbreviations

HAR: Human activity recognition; 3DSIFT: 3D Scale-Invariant Feature Transform; BOF: Bag-of-features; SVM: Support vector machine; TOP: Three orthogonal planes; SHMM: Simplex Hidden Markov Model; Fast-SHMM: Fast Simplex Hidden Markov Model; GF-OF: Gunner Farneback's dense optical flow; DBN: Deep Belief Network

Acknowledgements

Not applicable.

Funding

Not applicable, this study received no funding from any resource.

Availability of data and materials

All relevant data concerning this work are available from the corresponding author upon request. Datasets used in training and testing processes are public datasets.

Authors' contributions

SK (corresponding author) is the major contributor to this research; she conceived and designed the proposed approach, conducted the experiments, collected the results, analyzed them, and wrote the manuscript. MH and MF contributed in supervising the research and reviewing the manuscript. All authors read and approved the final manuscript.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 2 November 2021 Accepted: 27 April 2022
Published online: 21 May 2022

References

1. Sebe N, Cohen I, Garg A, Huang TS (2005) Machine learning in computer Vision vol. 29. SSBM, Berlin
2. Beddiar DR, Nini B, Sabokrou M, Hadid A (2020) Vision-based human activity recognition: a survey. Multimed Tools Appl 79(41):30509–30555
3. Zheng W-S, Gong S, Xiang T (2011) Person re-identification by probabilistic relative distance comparison. In: CVPR 2011. IEEE, New York. pp 649–656
4. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. arXiv preprint arXiv:1506.04214
5. Song Y, Li C, Wang Y (2017) Pixel-wise object tracking. arXiv preprint arXiv:1711.07377
6. Ren M, Zemel RS (2017) End-to-end instance segmentation with recurrent attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 6656–6664
7. Majd M, Safabakhsh R (2019) A motion-aware convlstm network for action recognition. Appl Intell 49(7):2515–2521
8. Liu T, Xu M, Wang Z (2019) Removing rain in videos: a large-scale database and a two-stream convlstm approach. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). IEEE, New York. pp 664–669
9. Alom MZ, Hasan M, Yakopcic C, Taha TM, Asari VK (2020) Improved inception-residual convolutional neural network for object recognition. Neural Comput & Applic 32(1):279–293
10. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local svm approach. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 3. IEEE, New York. pp 32–36
11. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2. IEEE, New York. pp 1395–1402
12. Rodriguez MD, Ahmed J, Shah M (2008) Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 1–8
13. Soomro K, Zamir AR (2014) Action recognition in realistic sports videos. In: Computer Vision in Sports. Springer, New York. pp 181–208
14. Jalal A, Lee S, Kim JT, Kim T-S (2012) Human activity recognition via the features of labeled depth body parts. In: International Conference on Smart Homes and Health Telematics. Springer, New York. pp 246–249
15. Jalal A, Kim Y, Kamal S, Farooq A, Kim D (2015) Human daily activity recognition with joints plus body features representation using kinect sensor. In: 2015 International Conference on Informatics, Electronics & Vision (ICIEV). IEEE, New York. pp 1–6
16. Kumar SS, John M (2016) Human activity recognition using optical flow based feature set. In: 2016 IEEE International Carnahan Conference on Security Technology (ICCST). IEEE, New York. pp 1–5
17. Niu F, Abdel-Mottaleb M (2004) View-invariant human activity recognition based on shape and motion features. In: IEEE Sixth International Symposium on Multimedia Software Engineering. IEEE, New York. pp 546–556
18. Althloothi S, Mahoor MH, Zhang X, Voyles RM (2014) Human activity recognition using multi-features and multiple kernel learning. Pattern Recog 47(5):1800–1812
19. Nazir S, Yousaf MH, Velastin SA (2018) Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition. Comput Electr Eng 72:660–669

20. Nadeem A, Jalal A, Kim K (2020) Human actions tracking and recognition based on body parts detection via artificial neural network. In: 2020 3rd International Conference on Advancements in Computational Sciences (ICACS). IEEE, New York. pp 1–6
21. Robertson N, Reid I (2006) A general method for human activity recognition in video. Comput Vis Image Underst 104(2-3):232–248
22. De P, Chatterjee A, Rakshit A (2017) Recognition of human behavior for assisted living using dictionary learning approach. IEEE Sensors J 18(6):2434–2441
23. Khaire P, Kumar P, Imran J (2018) Combining cnn streams of rgb-d and skeletal data for human activity recognition. Pattern Recogn Lett 115:107–116
24. Qi M, Qin J, Li A, Wang Y, Luo J, Van Gool L (2018) stagnet: an attentive semantic rnn for group activity recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer, New York. pp 101–117
25. Zaki Zadeh M, Babu AR, Jaiswal A, Makedon F (2020) Self-supervised human activity recognition by augmenting generative adversarial networks. arXiv e-prints
26. Singh R, Khurana R, Kushwaha AKS, Srivastava R (2020) A dual stream model for activity recognition: exploiting residual-cnn with transfer learning. Comput Methods Biomech Biomed Eng: Imaging Vis 9:1–11
27. Yuki Y, Nozaki J, Hiroi K, Kaji K, Kawaguchi N (2018) Activity recognition using dual-convlstm extracting local and global features for shl recognition challenge. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers. ACM, New York. pp 1643–1651
28. Kwon S, et al (2020) Clstm: Deep feature-based speech emotion recognition using the hierarchical convlstm network. Mathematics 8(12):2133
29. Aly S, Sayed A (2019) Human action recognition using bag of global and local zernike moment features. Multimed Tools Appl 78(17):24923–24953
30. Ramya P, Rajeswari R (2021) Human action recognition using distance transform and entropy based features. Multimed Tools Appl 80(6):8147–8173
31. Arunnehru J, Chamundeeswari G, Bharathi SP (2018) Human action recognition using 3d convolutional neural networks with 3d motion cuboids in surveillance videos. Procedia Comput Sci 133:471–477
32. Abdellaoui M, Douik A (2020) Human action recognition in video sequences using deep belief networks. Traitement Sig 37(1):37–44
33. Naseeb C, Saeedi BA (2020) Activity recognition for locomotion and transportation dataset using deep learning. In: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers. ACM, New York. pp 329–334
34. Han Y, Zhang P, Zhuo T, Huang W, Zhang Y (2018) Going deeper with two-stream convnets for action recognition in video surveillance. Pattern Recogn Lett 107:83–90
35. Zhang K, Zhang L (2018) Extracting hierarchical spatial and temporal features for human action recognition. Multimed Tools Appl 77(13):16053–16068
36. Abdelbaky A, Aly S (2020) Human action recognition based on simple deep convolution network pcanet. In: 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE). IEEE, New York. pp 257–262
37. Abdelbaky A, Aly S (2020) Human action recognition using short-time motion energy template images and pcanet features. Neural Comput & Applic 32:1–14
38. Abdelbaky A, Aly S (2021) Human action recognition using three orthogonal planes with unsupervised deep convolutional neural network. Multimed Tools Appl 80(13):20019–20043
39. Abdelbaky A, Aly S (2021) Two-stream spatiotemporal feature fusion for human action recognition. Vis Comput 37(7):1821–1835
40. Rodriguez M, Orrite C, Medrano C, Makris D (2016) One-shot learning of human activity with an map adapted gmm and simplex-hmm. IEEE Trans Cybern 47(7):1769–1780
41. Rodriguez M, Orrite C, Medrano C, Makris D (2017) Fast simplex-hmm for one-shot learning activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE, New York. pp 41–48
42. Haddad M, Ghassab VK, Najar F, Bouguila N (2021) A statistical framework for few-shot action recognition. Multimed Tools Appl 80:1–16
43. Singh A, Chakraborty O, Varshney A, Panda R, Feris R, Saenko K, Das A (2021) Semi-supervised action recognition with temporal contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 10389–10399
44. Jing L, Parag T, Wu Z, Tian Y, Wang H (2021) Videossl semi-supervised learning for video classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. IEEE, New York. pp 1110–1119
45. Srivastava N, Mansimov E, Salakhudinov R (2015) Unsupervised learning of video representations using lstms. In: International Conference on Machine Learning. PMLR, New York. pp 843–852
46. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 3367–3375
47. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, Granada. pp 5-14
48. Krizhevsky A, Hinton G, et al. (2009) Learning multiple layers of features from tiny images
49. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
50. Alom MZ, Hasan M, Yakopcic C, Taha TM (2017) Inception recurrent convolutional neural network for object recognition. arXiv preprint arXiv:1704.07709
51. Wei H, Zhou H, Sankaranarayanan J, Sengupta S, Samet H (2018) Residual convolutional lstm for tweet count prediction. In: Companion Proceedings of the The Web Conference 2018. IW3C2, Republic and Canton of Geneva. pp 1309–1316

52. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 1–9
53. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 770–778

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.