Journal of Artificial Intelligence and Technology, 2023, 3, 100-107
https://doi.org/10.37965/jait.2023.0163

ISTP
RESEARCH ARTICLE

Human Activity Recognition in a Realistic and Multiview Environment Based on Two-Dimensional Convolutional Neural Network

Ashish Khare,[1] Arati Kushwaha,[1] and Om Prakash[2]
[1]Department of Electronics and Communication, University of Allahabad, Allahabad, India
[2]Department of Computer Science, HNB Garhwal University, Srinagar, India

(Received 21 February 2023; Revised 02 April 2023; Accepted 02 April 2023; Published online 09 May 2023)

Abstract: Recognition of human activity based on convolutional neural network (CNN) has received the interest of researchers in recent years due to its significant improvement in accuracy. A large number of algorithms based on the deep learning approach have been proposed for activity recognition purpose. However, with the increasing advancements in technologies having limited computational resources, it needs to design an efficient deep learning-based approaches with improved utilization of computational resources. This paper presents a simple and efficient 2-dimensional CNN (2-D CNN) architecture with very small-size convolutional kernel for human activity recognition. The merit of the proposed CNN architecture over standard deep learning architectures is fewer trainable parameters and lesser memory requirement which enables it to train the proposed CNN architecture on low GPU memory-based devices and also works well with smaller as well as larger size datasets. The proposed approach consists of mainly four stages: namely (1) creation of dataset and data augmentation, (2) designing 2-D CNN architecture, (3) the proposed 2-D CNN architecture trained from scratch up to optimum stage, and (4) evaluation of the trained 2-D CNN architecture. To illustrate the effectiveness of the proposed architecture several extensive experiments are conducted on three publicly available datasets, namely IXMAS, YouTube, and UCF101 dataset. The results of the proposed method and its comparison with other state-of-the-art methods demonstrate the usefulness of the proposed method.

Keywords: computational resources; convolutional neural network; GPU memory; human activity recognition; softmax classifier; training parameters

I. INTRODUCTION

In the current era of research and rapid developments in computer vision applications, video-based human activity recognition is considered as one of the most popular research fields. Additionally, a human activity recognition system is beneficial for society in the sense that it can automatically detect human activities in videos or sensor inputs that occur in their day-to-day routine [1]. Human activity recognition plays an important role in many applications, including healthcare, video surveillance, driving safety, sports applications [1,2], etc. Although several studies have been conducted by computer vision scientists in this field and achieved success to a certain extent, but human activity recognition is still a challenging problem due to real-time processing, large intra-class difference, fuzzy boundary between classes, etc. Still, people keep on working to explore new technologies for activity recognition in improving accuracy, reducing computational resources, and developing a simplified model. Most of the previous works were based on handcrafted feature extraction-based techniques for activity recognition. Since real-world scenes are complex and have a range of varying information, handcrafted feature descriptors can grab only abstract level of information which cannot truly represent each activity class uniquely [2–4].

Inspired from the success of deep learning-based methods in a number of computer vision applications [5–7], several CNN-based human activity recognition methods were proposed. In [8], Yilmaz *et al.* designed a deep neural network architecture for action recognition and used a genetic algorithm for optimizing the proposed network. In [9], Muhammad *et al.* proposed an application for surveillance data based on the fusion of deep learning feature and handcrafted feature. For the extraction of deep learning features, they used pretrained VGG-19 deep learning architecture. In [10], Jaouedi *et al.* used the Gaussian mixture model and Kalman filter to extract the moving objects. The bounding box of the extracted moving object is further processed by a gated recurrent neural network for human action recognition. Leong *et al.* [11] have proposed a novel semi-convolutional neural network (CNN) deep learning architecture for human action recognition for video datasets. They evaluated their proposed architecture on UCF-101 dataset. Yang *et al.* [12] proposed an asymmetric 3D CNN architecture for video-based action recognition system. They used micronets in the construction of 3D CNN architecture to improve the performance of the architecture.

The methods discussed above focused on handcrafted and deep learning-based architecture. Further, few of them were based on the fusion of handcrafted features and deep learning features, which were complex and take more computation time in the prediction of activities. Further, in the presented techniques, 3-D CNN architectures required more computational resources and time [5]. However, the goal of a real-time application is always

Corresponding author: Om Prakash (e-mail: au.omprakash@gmail.com).

© The Author(s) 2023. This is an open access article published under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

to develop a efficient algorithm with less computational resources and improved performance.

However, in recent years, the CNN has outpaced traditional handcrafted feature-based machine learning techniques due to its automatic learning capability from the complex representations of large visual data. But, with the recent advancements in the mobile and embedded computing technologies that have limited computational resources, it encourages us to design deep learning-based efficient human activity recognition system for limited computational resources and memory that yields promising results [5]. Motivated from these facts, we propose a novel 2-D CNN for video-based human activity recognition for surveillance videos in a realistic environment. One of the major contributions of the proposed work is to design a lightweight 2-D CNN architecture with very small convolutional filters for human activity recognition from video data. The designed architecture have capability to be trained on the devices having very limited computational resources, and it has flexibility of training with small and large-size dataset. The proposed architecture is trained from scratch on low GPU memory with fewer trainable parameters as compared to the standard deep learning architectures (e.g., AlexNet and VGGNet) and achieved promising results. The considered proposed work consists of four main stages: 1. collect dataset and used data augmentation technique before training proposed CNN architecture in order to avoid overfitting problem; 2. design the 2-D CNN architecture; 3. train the proposed 2-D CNN architecture from scratch and train it upto optimum stage; and 4. evaluation the trained 2-D CNN architecture.

To prove the effectiveness of the proposed method, we have tested it on three different publically available datasets [13–15] and conducted several extensive experiments. The experimental results of the proposed method are compared with the result of existing standard deep learning architectures [6,16] and several existing state-of-the-art methods [4,9–13,17–30]. The obtained experimental results demonstrated the effectiveness of the proposed method.

The rest of the paper is organized as follows: section II consists of a detailed description of the proposed architecture and experimentation are given in section III. The results and discussion are presented in section IV and finally, section V presents the concluding remarks of the proposed work.

II. THE PROPOSED METHODOLOGY

The objective of this work is to present a framework of human activity recognition for realistic video-recorded from single or multiple cameras. In this work, we introduce a simple and lightweight 2-D CNN architecture which has the ability to learn complex invariant features from given input frame sequences. The proposed approach consists of four main stages:

i Data acquisition, which includes gathering video datasets followed by preprocessing steps before feeding these frame sequences into the network for training.

ii Design an efficient and simple 2-D CNN architecture.

iii Finally, we train the proposed 2-D CNN architecture from scratch till it is converging.

iv Used softmax classifier for evaluation of the trained 2-D CNN architecture.

A. PROPOSED ARCHITECTURE

Depending on the applications, the selection of the optimal deep learning architecture is a challenging task. While designing network architecture, we aim to design a simple and optimized network that learns unique and discriminative patterns from input data with fewer computational resources [5]. In recent years, a number of deep learning models such as AlexNet, VGGNet (e.g., VGG-16), GoogLeNet, etc. have been applied for image classification and have achieved good accuracy. VGG-16 is a 16-layer CNN, and it has remarkable feature extraction ability and achieves great success in image classification because it is deeper than the AlexNet, has more distinct feature representation, and has simpler and compact architecture than the GoogleNet like architectures and has better generalization ability [5]. Therefore, it has been used in a number of applications [6]. Therefore, motivated from it, in order to achieve a good trade-off between complexity and accuracy, we have designed a novel 2-D CNN architecture which is deeper like VGG-16 along but with less number of convolutional filters to reduce computational cost and time than VGG-16 and can be trained on low GPU memory for human activity recognition for realistic videos. We have opted simple and effective architecture which can also be trained on low GPU memory with fewer operational resources.

B. ARCHITECTURAL DETAILS

The proposed architecture for human activity recognition is shown in Fig. 1. The proposed architecture is consisting of 10 convolution layers (conv2-D_1 – conv2-D_10), four max-pooling layers (M1-M4), and three fully connected layers. Each of the hidden layers are equipped with the rectified linear unit (ReLU) activation function to increase the nonlinearity in the network. ReLU increases the nonlinear transformation to the input feature map of each layer, and it makes the decision function more discriminative and speeds up the training process [10,12]. All the convolution layers are processed by the kernel of size 3×3 and stride 1×1 and max-pooling uses 2×2 window size and stride (2, 2). Input to the proposed architecture is an RGB image of size (128×128). We have considered a very small convolution kernel (3×3) because it is sufficient to grab unique discriminative features from very small to that of large-size objects and has found improved performance in image classification applications [6]. A detailed description of kernel size, number of convolution filters, and the size of convolutional feature maps of the proposed architecture are given in Table I.

After the convolution operation, each feature map is processed by 1-pixel zero padding to keep the constant outcome of each convolution layer. Dropout is used after the last convolution layer to avoid overfitting. A stack of convolution layers which has different depths in each layer followed by three fully connected layers. The first two fully connected layers have 128 channels, and the third one has the number of channels equal to the number of activity categories, that is, one for each class. The last layer is followed by a softmax classifier to compute the class score of each activity category. The considered architecture is trained from scratch using publically available benchmark datasets until the network keeps on converging. A detailed description of the proposed architecture is given in a subsequent paragraph.

During the training network, we first feed input frame sequences of size $128 \times 128 \times 3$ in the first convolution layer (Conv2-D_1). In the first convolution layer, each input sequence is processed by 32 convolution kernels of size 3×3 with different random weights and stride 1. Mathematically Conve2-D_1 can be defined as:

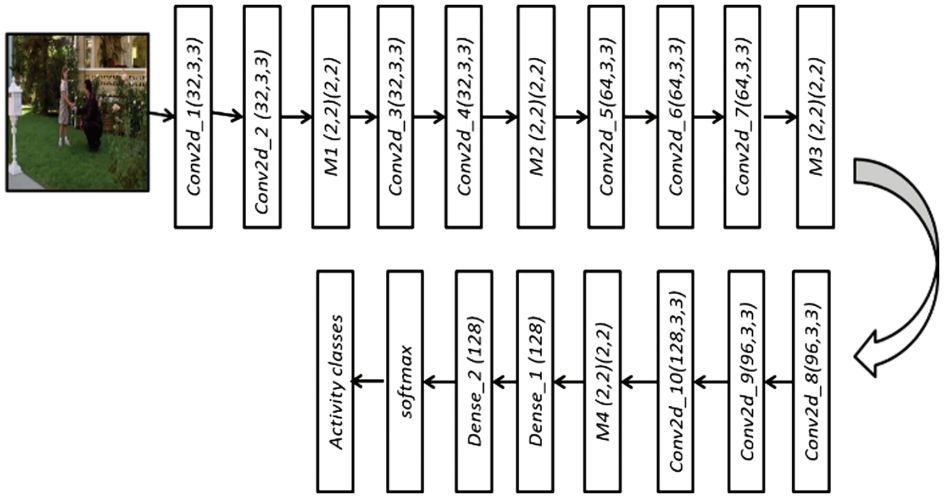

Fig. 1. The proposed 2-D CNN architecture for human activity recognition.

Table I. The architecture detail of feature map size, convolution kernel size, and number of convolution kernel used in the proposed architecture

Block	Kernel size	Kernel number	Feature map
Conv2 D_1	3×3	32	$32\times128\times128$
Conv2-D_2	3×3	32	$32\times128\times128$
M1	2×2	–	$32\times64\times64$
Conv2-D_3	3×3	32	$32\times64\times64$
Conv2-D_4	3×3	32	$32\times64\times64$
M2	2×2	–	$32\times32\times32$
Conv2-D_5	3×3	64	$64\times32\times32$
Conv2-D_6	3×3	64	$64\times32\times32$
Conv2-D_7	3×3	64	$64\times32\times32$
M3	2×2	–	$64\times16\times16$
Conv2-D_8	3×3	96	$96\times16\times16$
Conv2-D_9	3×3	96	$96\times16\times16$
Conv2-D_10	3×3	128	$128\times16\times16$
M4	2×2	–	$128\times8\times8$

$$\lambda^1 = f(w^1 \times X_i + b^1) \tag{1}$$

where w^1 represents weight matrix of first convolution layer with size $3\times3\times32$, b^l is bias vector of size $1\times1\times32$, x_i represents i^{th} training sample of j^{th} activity category of size $128\times128\times3$, and *f(.)* denotes the ReLU activation function which process feature maps to introduce the nonlinearity in the network and is mathematically defined as:

$$\mathrm{Re}LU = \max(\lambda,0) \tag{2}$$

Since the ReLU layer has no parameters. Therefore, in this layer, no learning is taking place. The output of the first convolution layer (feature map of size $128\times128\times32$) is utilized as input for the second convolution layer (Conv2-D_2) and processes it with 32, 3×3 convolution kernels and stride 1. The conv2-D_2 gives 32 feature map which is again processed by activation function that gives again feature map of size $128\times128\times32$. Mathematically, it can be formulated as follows:

$$\lambda^2 = f(w^2 \times \lambda^1 + b^2) \tag{3}$$

The obtained output feature map by conv2-D_2 is further processed by max-pooling layer (M1) with a window size 2×2 and stride [2,2] which outputs a feature map of size $64\times64\times32$.

The reason behind using the stack of two layers followed by max-pooling layer is that two convolution layer with kernel size 3×3 and stride 1 gives same effective receptive fields as one convolution layer with 5×5 convolution kernel. Therefore, using two convolution layers with kernel size 3×3 instead of one convolution layer with kernel size 5×5 is advantageous because it increases network depth and also introduces more nonlinearity into the network by using two ReLU operations one with each convolution layer and also reduces the number of learnable parameters, that is, computational resources (i.e., two stacks of convolution layer with 3×3 with C channels needs $2\times(32\times C2) = 18C2$ parameters, whereas at the same time single convolution layer with 5×5 kernel size needs $52\times C2 = 25C2$ parameters).

The obtained feature map from max-pooling layer M1 is further processed by the stack of two convolution layers (conv2-D_3 and conv2-D_4) with 32, convolution kernels of size 3×3 and $1\times1\times32$ bias, and convolutional layer results are followed by activation layer ReLU which gives $64\times64\times32$ feature maps. Mathematically, it can be represented as follows:

$$\lambda^3 = f(w^3 \times \lambda^2 + b^3) \tag{4}$$

$$\lambda^4 = f(w^4 \times \lambda^3 + b^4) \tag{5}$$

The obtained feature map ($64\times64\times32$) is then again processed by max-pooling layer (M2) with window size (2×2) stride [2,2] that gives a feature map of size $32\times32\times32$. The output of the max-pooling layer M2 is utilized as input of convolution layer 5. Therefore, the obtained feature map by layer M2 ($32\times32\times32$) is processed by the stack of three convolution layers (conv2-D_5, conv2-D_6, and conv_2-D_7) with 64, 3×3 convolution kernels and bias $1\times1\times64$. It can be mathematically represented as follows:

$$\lambda^5 = f(w^5 \times \lambda^4 + b^5) \quad (6)$$

$$\lambda^6 = f(w^6 \times \lambda^5 + b^6) \quad (7)$$

$$\lambda^7 = f(w^7 \times \lambda^6 + b^7) \quad (8)$$

The obtained feature map after convolution layer 7 (conv2-D_7) is processed by max-pooling layer M3, and this gives feature map of size $16 \times 16 \times 64$. These feature maps are again processed by the stack of three convolution layers (conv2-D_8, conv2-D_9, and conv2-D_10) with 96, 96, and 128, 3×3 convolution kernel which results in feature map of size $16 \times 16 \times 128$. Mathematical conv2-D_10 can be represented as:

$$\lambda^{10} = f(w^{10} \times \lambda^9 + b^{10}) \quad (9)$$

The output of conv2-D_10 is again processed by max-pooling layer M4 which results in feature map of size $8 \times 8 \times 128$. The obtained result of M4 layer is flattened and processed by three fully connected layers (L = 1,2,3) in which two fully connected layers (FC) have 128 channels and the last fully connected layer has number of channels equal to the activity category of the taken dataset, that is, one channel for each activity. Mathematical feature vector computation at fully connected layer is as follows:

$$\lambda^{FC} = f(w^L \times \lambda^{10} + b^L) \quad (10)$$

where w^L and b^L represent weight vector and bias vector, respectively, for a fully connected layer.

In the last FC layer, we have used a softmax classifier to compute class scores, which allows us to interpret the output as a probability. Categorical cross-entropy loss is utilized to measure loss sometimes also called as error (cost) at the softmax layer.

Once we processed the input data to the network, we compute the loss (error) of the network using the predicted output at the last FC layer and their ground truth. Then the computed loss is backpropagated from the last layer to the first layer. The backpropagation algorithm provides us gradients of the error which is then utilized to update the learning parameters (weights and bias). In the training process with each epoch, we repeatedly compute gradients of the loss function and perform parameter updation using the above-mentioned procedure till the network is converging. Therefore, in this way we keep on training and updating learning parameters till we not reach the minimum error. The overall flow of the proposed method is shown in Fig. 2.

III. EXPERIMENTATION

In this section, we present experimental setups including implementation details, datasets, and evaluation criteria to measure the performance of the proposed 2-D CNN architecture.

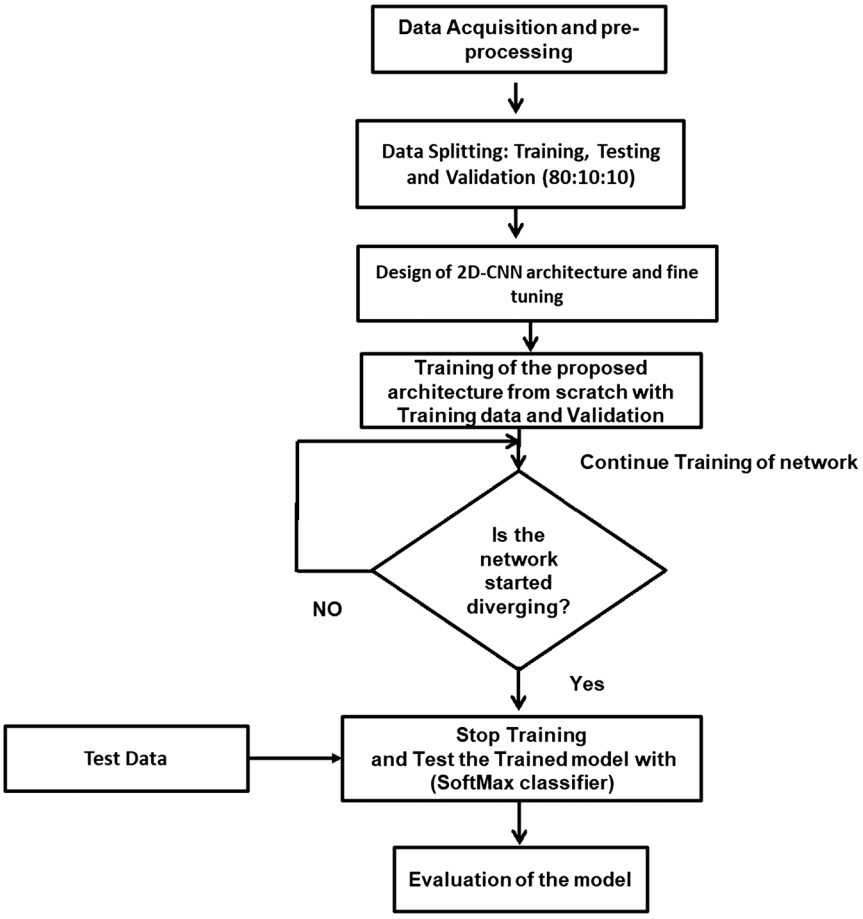

Fig. 2. The overall flow of the proposed method.

A. IMPLEMENTATION DETAILS

To determine the empirical justification of the proposed work, we performed several experiments on three different publically available video datasets [13–15]. These video datasets are firstly converted into frame sequences before feeding into network for training. For training and testing, both RGB frame sequences have been used. The frame sequences are first resized to 128×128 to reduce the computational resources and computation time, then we used data augmentation techniques such as rotation, translation, and zooming in order to expand the training dataset which resolves the limitation of smaller dataset size, and this makes the network to perform better generalization effect [31]. To implement the proposed CNN architecture, we used the Keras library and experimented on Nvidia P2000 GPU having Intel® Xeon® CPU E7-4809 processor. The implemented 2-D CNN architecture is trained from scratch using ADAM optimizer [32] with momentum 0.9 and 0.99 and 0.001 learning rate.

B. DATASET DESCRIPTION

To evaluate the proposed architecture, we have taken five publically available benchmark datasets, viz. IXMAS [13], YouTube [14], and UCF101 [15]. These datasets are briefly described as follows:

IXMAS Dataset: It is a publically available multi-view dataset in which videos were captured from different views of five cameras [13]. This dataset consists of low-resolution, 13 daily life activities performed by 11 actors. Its activity categories were do nothing, check watch, crossing arms, etc. This dataset was introduced by INRIA, France, in 2006.

YouTube Dataset: It is a publically available sports dataset which consists of total of 11 activity categories [14]. The videos are captured by 25 groups of individuals with more than four video clips in it in which more than 100 sample videos of each activity category. The video clips of each group have some common features such as similar background, same actor, and similar viewpoint, etc. These videos are captured in a realistic environment and have challenges like camera motion, the appearance of object and pose, object scale, etc.

UCF101 Dataset: The dataset UCF101 [15] consists of realistic user-uploaded video clips with a total of 13,320 video clips. These video clips are captured under varying illumination conditions, cluttered scenes, etc. It is one of the most challenging datasets of video-based human activity recognition system due to its large number of activity categories, a large number of video clips, and also the unconstrained nature of such video clips. This dataset consists of total 101 activity categories ranging from daily life activity to sports. This dataset can be further categorized into five groups: (1) Human–Object Interaction, (2) Body-Motion Only, (3) Human–Human Interaction, (4) Playing Musical Instruments, and (5) Sports.

C. EVALUATION CRITERIA

To authenticate the usefulness of the proposed architecture, we compared the proposed architecture with standard deep learning architectures, that is, AlexNet and VGGNet (VGG-16) in terms of fewer learnable parameters, per frame memory required (GPU memory) in a single pass on training, classification accuracy, and convergence rate (which is a measure of minimum epochs taken to reach test accuracy at optimum value) [33].

IV. RESULTS AND DISCUSSION

In this section, we present extensive experiments and their outcomes. Several experiments were conducted to evaluate the proposed architecture and its usefulness. We performed experiments on five publically available video datasets: IXMAS [13], YouTube [14], and UCF101 [15]. The results were critically analyzed with respect to other state-of-the-art methods.

A. EVALUATION OF THE PROPOSED 2-D CNN ARCHITECTURE

To evaluate the effectiveness of the proposed architecture, Firstly, we compared, the proposed architecture with standard deep learning architectures, that is, AlexNet [16] and VGGNet [6] (VGG-16) in terms of learnable parameters and memory required per frame and classification accuracy. Therefore, the proposed architecture and standard deep learning architectures [6,16] are trained from scratch on YouTube dataset [14]. The experimental results are given in Table II.

From Table II, one can see that the proposed architecture is giving comparable result (99.89% classification accuracy value) to standard deep learning architectures but requires very fewer learnable parameters for training, that is, 1.4 million only and lesser memory required, that is, 6.7 MB per frame which is smaller in comparison to AlexNet (25MB/frame) and VGGNet (70.8MB/frame). This demonstrates that the proposed architecture is computationally efficient and requires less computational resources; therefore, it is found suitable for devices having limited resources (like low GPU memory).

In addition, we also presented the learning curve in Fig. 3. The learning curves are drawn for both training versus validation accuracy and training versus validation loss.

From Fig. 3(a) and (b), one can observe that the curves are showing poor convergence as well as overfitting problems for AlexNet and VGGNet architectures, that is, more learnable parameters than samples for training which show the learning capacity of these networks are more than the data used for training. From Fig. 3(c), one can see that the proposed architecture shows good convergence and comparable results to the standard deep learning architectures (AlexNet and VGGNet) in lesser memory and learnable parameters. This is due to the proposed architecture design with a lesser number of neurons in each layer unlike AlexNet and deeper architecture design like VGGNet, with very few computational resources. The deeper structure enables this model to extract

Table II. Comparison of proposed architecture with standard deep learning architectures in terms of memory required for feature map and learnable parameters

Architectures	Learnable parameters (million)	Feature map (MB/frame)	Classification accuracy (%)
AlexNet	71	25.0	99.56
VGGNet	134	70.80	99.00
Proposed Architecture	1.4	6.70	99.89

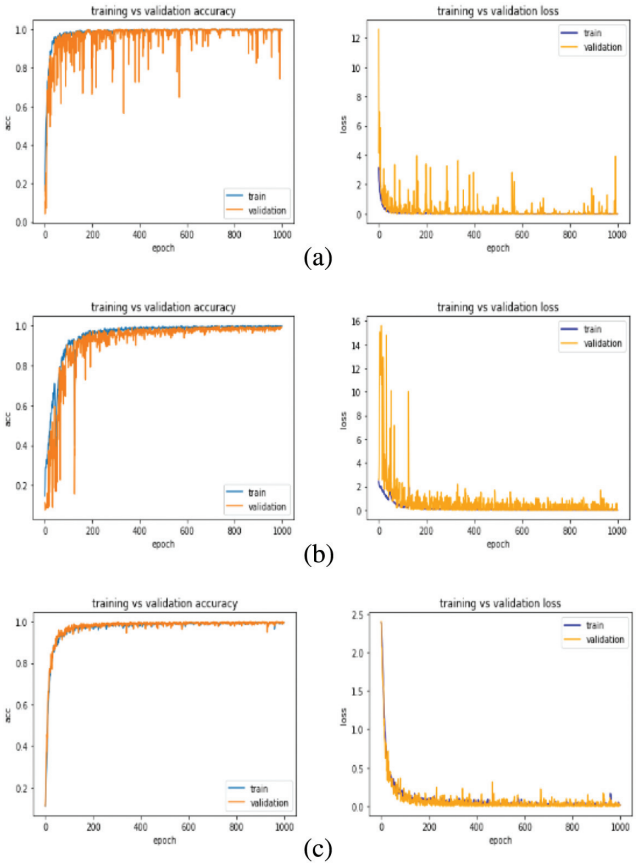

Fig. 3. Learning curves (first column – training versus validation accuracy and second column – training versus validation loss) for (a) AlexNet architecture, (b) VGGNet and (c) proposed architecture.

more abstract information and therefore represents each activity class uniquely.

We again experimented the proposed architecture with the RMSProp optimizer to see the role of the optimizer in the learning and accuracy of the proposed architecture with the same learning parameters and bath size as previous up to 1000 and 1500 epochs from scratch and achieved accuracy of 98.54% at 1000 epochs and 97.93 at 1500 epochs. The learning curve for the RMSProp optimizer of the proposed architecture is shown in Fig. 4.

From Figs. 3(c), 4(a), and (b), one can observe that the proposed architecture performs well with the ADAM optimizer than the RMSProp optimizer. This is due to the positive accumulated gradient used in parameter updating that sometimes kills the learning process due to a decrease in the effect of learning rate in RMSProp. This is the reason why we used ADAM [32] optimizer for further experimentation.

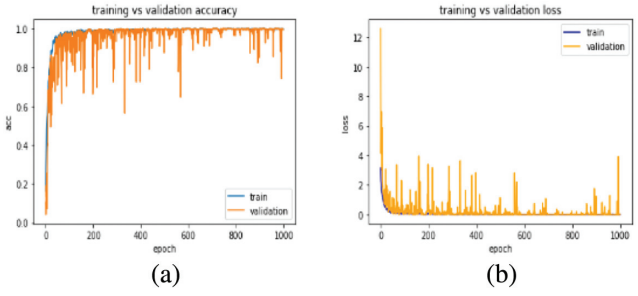

Fig. 4. Learning curve of proposed architecture on RMSProp optimizer (a) at 1000 epochs and (b) at 1500 epochs.

Table III. Classification accuracy on the dataset IXMAS [13]

Methods	Classification accuracy (%)
Kushwaha *et al.* [4]	99.50
Khan *et al.* [9]	95.20
Kim *et al.* [13]	77.20
Elharrouss *et al.* [18]	99.60
Saregano *et al.* [23]	89.75
Gnouma *et al.* [24]	92.18
Khan *et al.* [29]	99.60
Proposed Method	99.58

B. COMPARISON OF THE PROPOSED METHOD WITH OTHER EXISTING METHODS

We compared the experimental results of the proposed 2-D CNN architecture with other state-of-the-art methods to prove the effectiveness of the proposed architecture. First, we experimented the proposed architecture on IXMAS [13] dataset and achieved an accuracy 99.58% at 2000 epochs. The results on the IXMAS dataset for the proposed architecture and other methods used in comparison [4,9,13,17,18,23,24,29] are given in Table III.

From Table III, we can observe that the proposed methods achieve 99.58 % classification accuracy which is a result comparable to the methods proposed by Khan *et al.* [29], Elharrouss *et al.* [18], and Kushwaha *et al.* [17] in less computational budget. This is due to the extraction of more abstract information about the class of the data which enables the proposed architecture to learn the complementary and unique features of activities from the activity frame sequences of multi-view environment. Thus, it reveals that the proposed architecture is found suitable for multi-view camera environments.

In the next experiment, we used the YouTube dataset [14]. In this dataset, we trained the proposed architecture from scratch and achieved a classification accuracy 99.83% at 4000 epochs. The classification accuracy of the proposed architecture and methods used in comparison [9,20–22,26,27,29] is shown in Table IV.

From Table IV, it can be observed that the proposed method achieves 99.83% classification accuracy which is comparable to the result proposed by Khan *et al.* [29]. However, the method proposed by Khan *et al.* [29] achieves the highest classification accuracy (100%), but this method is more computationally complex than the

Table IV. Classification accuracy on YouTube [14] dataset

Methods	Classification accuracy (%)
Khan *et al.* [9]	99.40
Wang *et al.* [20]	98.76
Zebhi *et al.* [21]	93.40
Meng *et al.* [22]	89.70
Abdelbaky *et al.* [26]	81.40
Afza *et al.* [27]	94.50
Khan *et al.* [29]	100
Proposed Method	99.83

Table V. Classification accuracy on UCF101 dataset [15]

Methods	Classification accuracy (%)
Jaovedi *et al.* [10]	89.30
Leong *et al.* [11]	89.00
Yang *et al.* [12]	92.60
Li *et al.* [17]	95.90
Yu *et al.* [19]	91.40
Wang *et al.* [20]	84.10
Chaudhary *et al.* [25]	97.70
Zhang *et al.* [28]	93.72
Zhang *et al.* [30]	95.10
Proposed method	96.23

proposed method as discussed above. Therefore, the proposed method is found computationally efficient and better in terms of classification accuracy.

Finally, we conducted the experiments on the dataset UCF101 [15] which consists of 101 activity categories. To perform experiment over UCF101 [15], we trained the network from scratch up to 25k epochs and achieved optimum classification accuracy value 96.23% at 15k epochs. Again, the classification accuracy was calculated for the proposed method and other state-of-the-art methods [10–12,17,19,20,25,28,30] used in the comparison. These calculated accuracy values are given in Table V.

From Table V, we observed that the proposed architecture achieved 96.23% classification accuracy which is comparable to the other state-of-the-art methods. Here, we can see that the method proposed by Chaudhary *et al.* [25] resulted in better classification accuracy value 97.70%, but the method proposed by them is more computationally complex than the proposed architecture. Because, Chaudhary *et al.* [25] first construct depth images using frame sequences which are used for training the CNN training. Then it involves the extra computational burden of constructing the depth images that require more computation time and operational resources. Therefore, we again found that the proposed architecture is computationally efficient and achieves comparable result to the other state-of-the-art methods resulting in less computational resources and low GPU memory.

Thus, from the extensive experiments on different challenging datasets [13–15] and their results given in Tables II and Tables III–V, we observed that the proposed architecture 2-D CNN architecture is a simple and efficient that requires lesser learnable parameters and therefore requires less computational resources and, hence, less GPU memory for training than the standard CNN architectures (see in Table II). Fig. 3(c) also reveals that the proposed method is less prone to overfitting problems with small-size datasets. Therefore, the proposed method is found suitable for real-time applications having limited computational resources and sample data.

V. CONCLUSIONS

In this paper, we presented a deep learning-based approach for human activity recognition in a realistic and multi-view environment. We designed a simple and a computationally efficient, lightweight two-dimensional 2-D CNN with a very small-size convolutional kernel for activity recognition. The proposed architecture is found useful for range of challenging scenarios such as human–human interaction, human–object interaction, varying illumination, inter-class variation, low-resolution videos, etc. The proposed 2-D CNN is fine-tuned and trained from scratch using video frame sequences. The proposed method is also found less complex than the standard architectures, require fewer learnable parameters and memory per frame, which yields the requirement of fewer computational resources such as low GPU memory for training, and is found suitable for smaller size datasets as well. The proposed architecture is trained and tested over several publically available benchmark datasets [13–15]. The usefulness of the proposed method was analyzed by comparing its results with other existing methods. From the detailed analysis of experimental results, it has been found that the proposed architecture produces competitive results with high computational efficiency. From the experiments and its exhaustive analyses, it has been found that the proposed method performs well for high-level complex human activities recorded in a multi-view and realistic environment.

References

[1] J. Wang et al., "Deep learning for sensor-based activity recognition: a survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, 2019.

[2] A. Kushwaha et al., "Dense optical flow based background subtraction technique for object segmentation in moving camera environment," *IET Image Process.*, vol. 14, no. 14, pp. 3393–404, 2020.

[3] P. Srivastava and A. Khare, "Utilizing multiscale local binary pattern for content-based image retrieval," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 12377–12403, 2018.

[4] A. Kushwaha et al., "On integration of multiple features for human activity recognition in video sequences," *Multimedia Tools Appl.*, vol. 80, no. 21, pp. 32511–32538, 2021.

[5] C. Szegedy et al., "Going deeper with convolutions," In *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, IEEE, pp. 1–9, 2015.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, Sep. 4, 2014.

[7] R. Yamashita et al., "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[8] A. A. Yilmaz et al., "A novel action recognition framework based on deep-learning and genetic algorithms," *IEEE Access*, vol. 8, pp. 100631–100644, 2020.

[9] M. A. Khan et al., "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia Tools Appl.*, pp. 1–27, 2020.

[10] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 4, pp. 447–53, 2020.

[11] M. C. Leong et al., "Semi-CNN architecture for effective spatio-temporal learning in action recognition," *Appl. Sci.*, vol. 10, no. 2, p. 557, 2020.

[12] H. Yang et al., "Asymmetric 3d convolutional neural networks for action recognition," *Pattern Recognit.*, vol. 85, pp. 1–2, 2019.

[13] S. J. Kim et al., "View invariant action recognition using generalized 4D features," *Pattern Recognit. Lett.*, vol. 49, pp. 40–47, 2014.

[14] J. Liu et al., "Recognizing realistic actions from videos 'in the wild'," In *2009 IEEE Conf. Comput. Vision Pattern Recognit.*, IEEE, pp. 1996–2003, 2009.

[15] K. Soomro et al., "UCF101: a dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, pp. 1–7, Dec. 3, 2012.

[16] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] J. Li et al., "Spatio-temporal deformable 3d convnets with attention for action recognition," *Pattern Recognit.*, vol. 98, p. 107037, 2020.

[18] O. Elharrouss et al., "A combined multiple action recognition and summarization for surveillance video sequences," *Appl. Intell.*, vol. 51, no. 2, pp. 690–712, 2021.

[19] S. Yu et al., "Learning long-term temporal features with deep neural networks for human action recognition," *IEEE Access*, vol. 8, pp. 1840–1850, 2019.

[20] L. Wang et al., "Human action recognition by learning spatio-temporal features with deep neural networks," *IEEE Access*, vol. 6, pp. 17913–17922, 2018.

[21] S. Zebhi et al., "Video classification by fusing two-stream image template classification and pretrained network," *J. Electron. Imaging*, vol. 29, no. 5, 053011, 2020.

[22] B. Meng et al., "Human action recognition based on quaternion spatial-temporal convolutional neural network and LSTM in RGB videos," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 26901–26918, 2018.

[23] A. B. Sargano et al., "Human action recognition from multiple views based on view-invariant feature descriptor using support vector machines," *Appl. Sci.*, vol. 6, no. 10, p. 309, 2016.

[24] M. Gnouma et al., "Stacked sparse autoencoder and history of binary motion image for human activity recognition," *Multimedia Tools Appl.*, vol. 78, no. 2, pp. 2157–2179, 2019.

[25] S. Chaudhary and S. Murala, "Depth-based end-to-end deep network for human action recognition," *IET Comput. Vision*, vol. 13, no. 1, pp. 15–22, 2018.

[26] A. Abdelbaky et al., "Human action recognition using three orthogonal planes with unsupervised deep convolutional neural network," *Multimedia Tools Appl.*, vol. 80, no. 13, pp. 20019–20043, 2021.

[27] F. Afza et al., "A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection," *Image Vis Comput.*, vol. 106, p. 104090, 2021.

[28] L. Zhang et al., "Few-shot activity recognition with cross-modal memory network," *Pattern Recognit.*, vol. 108, p. 107348, 2020.

[29] M. A. Khan et al., "Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition," *Appl. Soft Comput.*, vol. 87, p. 105986, 2020.

[30] C. Zhang et al., "Hybrid handcrafted and learned feature framework for human action recognition," *Appl. Intell.*, vol. 52, pp. 1–7, 2022.

[31] S. Pang et al., "VGG16-T: a novel deep convolutional neural network with boosting to identify pathological type of lung cancer in early stage by CT images," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 771–780, 2020.

[32] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, pp. 1–15, Dec. 22, 2014.

[33] A. Kushwaha et al., "Micro-network-based deep convolutional neural network for human activity recognition from realistic and multi-view visual data," *Neural Comput. Appl.*, pp.1–21, 2023.