

# Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications

Amin Ullah<sup>a</sup>, Khan Muhammad<sup>b</sup>, Weiping Ding<sup>c</sup>, Vasile Palade<sup>d</sup>, Ijaz Ul Haq<sup>a</sup>, Sung Wook Baik<sup>a,\*</sup>

<sup>a</sup> Sejong University, Seoul, South Korea

<sup>b</sup> Department of Software, Sejong University, Seoul, South Korea

<sup>c</sup> Department of Computer Science and Technology, Nantong University, China

<sup>d</sup> Department of Environment and Computing at Coventry University, UK



## ARTICLE INFO

### Article history:

Received 1 March 2020

Received in revised form 5 January 2021

Accepted 6 January 2021

Available online 15 January 2021

### Keywords:

Artificial intelligence

Machine learning

Pattern recognition

IoT

Activity recognition

Video big data analytics

Deep learning

GRU

## ABSTRACT

Recognizing human activities has become a trend in smart surveillance that contains several challenges, such as performing effective analyses of huge video data streams, while maintaining low computational complexity, and performing this task in real-time. Current activity recognition techniques are using convolutional neural network (CNN) models with computationally complex classifiers, creating hurdles in obtaining quick responses for abnormal activities. To address these challenges in real-time surveillance, this paper proposes a lightweight deep learning-assisted framework for activity recognition. First, we detect a human in the surveillance stream using an effective CNN model, which is trained on two surveillance datasets. The detected individual is tracked throughout the video stream via an ultra-fast object tracker called the 'minimum output sum of squared error' (MOSSE). Next, for each tracked individual, pyramidal convolutional features are extracted from two consecutive frames using the efficient LiteFlowNet CNN. Finally, a novel deep skip connection gated recurrent unit (DS-GRU) is trained to learn the temporal changes in the sequence of frames for activity recognition. Experiments are conducted over five benchmark activity recognition datasets, and the results indicate the efficiency of the proposed technique for real-time surveillance applications compared to the state-of-the-art.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

In the development of smart cities, human activity recognition from visual data is one of the most challenging research areas, especially in surveillance applications. The abnormal activity recognition is very crucial for law enforcement agencies to prevent crime. The abnormal activities refer to the ones which are dangerous for human life or property such as accident, activities which damage properties, against the law, or criminal activities such as fight, stealing. The techniques developed in the initial research of activity recognition are evaluated using different datasets that contain activities performed by a single actor in controlled settings. However, the current research focus has shifted towards uncontrolled, realistic video datasets that pose greater challenges to activity recognition tasks, such as background clutter, inter- and intra-class variations, occlusion, pose changes, camera motion, etc. [1,2]. Current research also concerns pedestrian detection and tracking in complex surveillance scenarios for effective analysis. Human activity is captured in a

sequence of video frames: therefore, the recognition of human activity relies on both visual appearance and its fusion with motion dynamics in a sequence of frames [3]. Recently, CNNs have achieved remarkable performance in image classification and object detection tasks. As CNNs process a single image at a time, they cannot be used directly to classify visual data in a time series. Some researchers introduced 3D CNNs that can also capture the spatio-temporal information in visual data; however, they can only process 10 to 15 frames effectively. They cannot efficiently recognize lengthy activities because of the exponential increase in time complexity caused by increasing the length in the third dimension.

The next challenging problem is modeling the temporal variations in the video; this is especially difficult if the recognition is performed online or in real-time surveillance systems. For instance, traditional approaches based on trajectories are highly dependent on optical flow-based hand-crafted features [4,5]. Similarly, some of the deep end-to-end multi-stream approaches [6, 7] use multiple 2D networks corresponding to the optical flow and the appearance, respectively. The performance of these methods is good but extracting the optical flow features makes them computationally expensive; therefore, they face challenges when

\* Corresponding author.

E-mail address: [sbaik@sejong.ac.kr](mailto:sbaik@sejong.ac.kr) (S.W. Baik).

it comes to largescale datasets and real-time surveillance systems. Later, the problem of expensive computation of the spatio-temporal features is solved by utilizing 3D CNNs. Recently, a variety of 3D CNNs with different architectures have been deployed for action recognition, such as the two-stream 3D ConvNet [8], pseudo-3D CNNs [9], MiCT-Net [10], etc. These 3D CNNs have the ability to directly extract spatio-temporal features, which significantly improves the performance in terms of classification and time complexity. However, some of these hybrid methods also train sequential learning models such as RNN [11] and LSTM [12] to improve the classification accuracy for action recognition, which again results in expensive computations.

The aforementioned methods involved numerous processing tasks, because first, they extracted the spatial features using CNNs or optical flow models, and then they fused these features to their LSTM for sequence learning. Generally, an LSTM contains input, output, forget gates, and a memory cell, which keeps the earlier information of the sequence. Therefore, the idea of a deep or multi-layer LSTM introduced earlier is computationally more expensive, because each layer of the deep LSTM has its gates and memory cell. These high numbers of computations make the activity recognition system ineffective for real-time surveillance applications. To tackle the problem of high computational complexity for activity sequence learning, this paper proposes a deep recurrent skip connection (DS-GRU) architecture, which is a variant of an RNN that contains only two gates with no memory cell. This makes the proposed technique capable of learning long-term sequences efficiently. The main contributions of this work are listed below:

1. An important step for surveillance activity recognition is to detect, localize, and track each individual throughout the video stream. This task is not feasible with object detectors that are trained on general categories of data. For this purpose, we fine-tuned a lightweight CNN model for human detection with new data and enabled it to work in a changing surveillance environment. Next, we utilized an ultra-fast target tracker, which can process more than 250 frames per second, to track a human in a video [13].
2. Sequential feature extraction from video data has a high computational cost due to its high dimensionality. We investigated an efficient LiteFlowNet CNN model, which is originally designed for optical flow detection. This model is 30 times smaller and 1.36 times faster than state-of-the-art optical flow CNN models [14,15]. We extracted pyramidal convolutional features in a novel way from its intermediate layers to capture motion between consecutive frames.
3. LSTM is the most popular neural network for learning time series data. However, due to its complex gated structure and memory units, it takes an intolerable amount of time to process sequential data for real-time scenarios. We presented an alternative, a DS-GRU, which is 1.4 times faster than a deep LSTM network with similar accuracy.
4. We proposed a lightweight activity recognition model, with a size of 48 MB, that can be easily embedded into vision sensors for efficient surveillance. It is superior to state-of-the-art methods, its effectiveness is verified from experiments, and it is much smaller than the C3D model of size 321 MB, the MiCT-Net of size 221 MB, and the ML-LSTM of size 193 MB.

The rest of the paper is organized as follows: all technical details of the proposed technique are discussed in Section 2. The experimental evaluation of the technique and a discussion of the results are given in Section 3. Section 4 summarizes the key findings of this article and recommends future research directions.

## 2. Related work on activity recognition

Human activity recognition in videos got tremendous attention in the computer vision community after the advancements in deep learning for image classification and object detection tasks. There are comprehensive surveys on traditional and deep learning-based human action and activity recognition methods in the literature [16,17]. In this paper, we summarized some of the important approaches for video sequential data learning using deep learning since video data analytics required both spatial and temporal features for its analysis. The mainstream methods mostly extract frame-level deep CNN features from pretrained models, followed by some classification or sequence learning technique for activity recognition. In such techniques, recurrent neural networks (RNNs) and their variant known as 'long short-term memory' networks (LSTMs) are dominant due to their ability to perform sequential modeling, making them suitable for learning long-term temporal dynamics in videos. For instance, Li et al. [18] extracted C3D features from a fixed-length video sequence using a sliding window approach and generated cubes in the spatial direction for the whole video. They introduced a part selection mechanism to select only informative cubes in the spatial direction. The selected cubes are then fed into a multi-layer LSTM network for sequence learning, where the SoftMax layer yields the predictions. Gammulle et al. [19] proposed a deep fusion framework that combined salient spatial features and their temporal relationships using two-stream LSTM network for action recognition.

In order to deal with non-stationary problems in video sequences, Sun et al. [20] introduced the Lattice-LSTM ( $L^2$ STM) in which, for each spatial location, the hidden state transitions of the memory cell learn independently. They jointly trained their  $L^2$ STM model with RGB and optical flow information, which accurately control the dynamics of the memory cell. They claim that their model can learn long-term motion dynamics without significantly increasing the model complexity. Shugao et al. [21] proposed a formulation for the ranking loss on the discriminative margin of the LSTM to accurately learn the recognition model. They extracted deep features from each frame of the video sequence using VGG19 and then fed these features into an LSTM for activity recognition. They mainly focused on the training loss and proposed a novel ranking loss, which is used together with the classification loss to train the LSTM model. Another approach in [22] proposed a VideoLSTM, which is an end-to-end sequence learning architecture. First, they extracted 2-D spatial features and then fed them into a convolutional LSTM network to learn spatio-temporal features for action recognition. Furthermore, they also localize the actions in the video sequence by utilizing the motion contents in the sequence. Two recent methods proposed by Ullah et al. [1,23] investigated deep features of the AlexNet CNN model, followed by a deep bi-directional LSTM and convolutional features from the FlowNet2 model, followed by a multi-layer LSTM for action and activity recognition, respectively.

Recently, Majd and Safabakhsh [12] proposed an extended version of LSTM called the Correlational Convolutional LSTM ( $C^2$ LSTM), which perceives motion, spatial features, and time dependencies all in one unit. This method has the ability to predict the starting and ending components of a whole activity. For example, for the class eating, it can predict 'passing spoon before eating'. Similarly, Kuehne et al. [24] tackled the problem of weakly supervised learning of human actions by proposing a hybrid RNN-HMM approach. Qi et al. [25] proposed stagNet, a novel attention semantic RNN for human group activity recognition, by combining the spatio-temporal attention mechanism and semantic graph modeling. Additionally, this method can compute body-region attention and a global-parts features pooling strategy for individual action recognition.

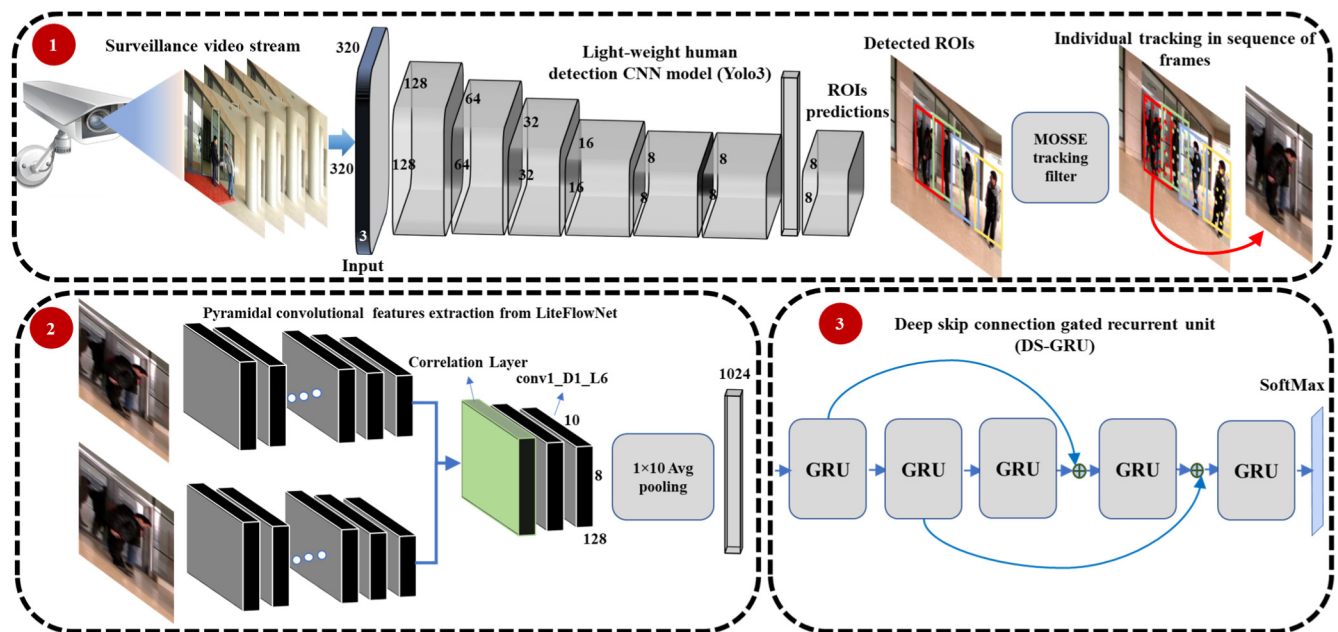


Fig. 1. The proposed activity recognition framework for surveillance applications.

### 3. Proposed methodology

In this section, the core phases of the proposed technique are discussed in detail. The detailed framework of the proposed technique, which is divided into three core phases, is given in Fig. 1. In the first phase, we detect pedestrians in the surveillance video using the lightweight CNN model. Each individual is then tracked using the MOSSE tracker, in order to capture its activity sequences. In the second phase, pyramidal optical flow features are extracted from two consecutive frames. Finally, sequential patterns are learned using the proposed DS-GRU network for activity recognition.

#### 3.1. Training a human detection model

Human detection is an important step for activity recognition, and there exist several methods for this task. However, their efficiency and effectiveness do not meet the standards of activity recognition in surveillance applications. The pretrained YOLOv3 [26] object detection model is very efficient, and it can also detect humans. However, it is trained on general categories' datasets, which contains many other objects along with humans that are not related to surveillance data. Therefore, we have fine-tuned the YOLOv3 object detection model on two large scale pedestrian detection datasets of surveillance data, including the SPID [27] and Caltech [28] datasets. The YOLOv3 model trained for a specific object (pedestrian) is more powerful than one that is trained on general categories data. This helps the model to detect humans in challenging surveillance data with different poses and scales in varying illuminations scenarios because we have trained it on two combined datasets. We have fine-tuned the YOLOv3 with Darknet-53 on the backend as a feature extractor for an input size of  $320 \times 320$ . Darknet-53 contains small consecutive convolutional filters of  $3 \times 3$  and  $1 \times 1$ , which help to detect humans of different scales even for large distances. It uses logistic regression to detect objects and their bounding box confidence scores. The key reason for using Darknet-53 as a backend model is its efficiency. Redmon et al. [26] experimentally prove under the same processing settings that Darknet-53 is more efficient than ResNet-152, Darknet-19, ResNet-101, etc., as shown in Table 1. Due to its high speed and a smaller number of

Table 1

A comparison of various backbone models for object detectors using frame per second (FPS), number of operations in billions (Ops Bn), number of floating operations in billions per second (FLOpBn/s), Top-1, and Top-5 accuracies.

Model	FPS	Ops Bn	FLOpBn/s	Top-1	Top-5
ResNet-101 [29]	53	19.7	1039	77.1	93.7
ResNet-152 [29]	37	29.4	1090	77.6	93.8
Darknet-19 [30]	171	7.29	1246	74.1	91.8
Darknet-53 [31]	<b>78</b>	<b>18.7</b>	<b>1457</b>	<b>77.2</b>	<b>93.8</b>

floating-point operations, Darknet-53 performs better than state-of-the-art methods. Table 1 shows the results of ResNet-152 are similar to Darknet-53, but it is two times slower. ResNet-101 is 1.5 times slower than Darknet-53, and Darknet-53 performs better than ResNet-101. Darknet-53 also achieves the maximum per second estimated floating-point operations. We achieved 32.56 mean average precision (mAP) for the fine-tuned model on the combined dataset. It takes only 22 ms to process each frame [26], which makes it a good solution to our proposed problem of activity recognition in real-time surveillance applications. The reader can refer to [26] for more details about YOLOv3.

#### 3.2. Tracking humans for sequence analysis

Activity recognition is the process of analyzing sequential actions performed by humans in video streams. In surveillance environments, a number of activities are performed at a time, and each individual activity has its own significance. For this purpose, after the detection of a pedestrian, we need to track him in the video stream to precisely capture his motion sequences for further analysis. Recently, Ullah et al. [1] utilized the activations of the convolutional feature maps of a pretrained MobileNet CNN model to extract the salient regions where humans perform an activity. Dai et al. [32] presented a temporal context network (TCN) for human activity localization in the video stream; their network architecture is similar to Faster-RCNN. However, the exploitation of the deep architecture as a pre-processing step for activity recognition makes the method inefficient for real-time surveillance applications. Therefore, we have utilized an ultra-fast visual object tracker known as the MOSSE [13] tracking filter





Fig. 2. Results of localized activity using the MOSSE tracker in a surveillance environment.

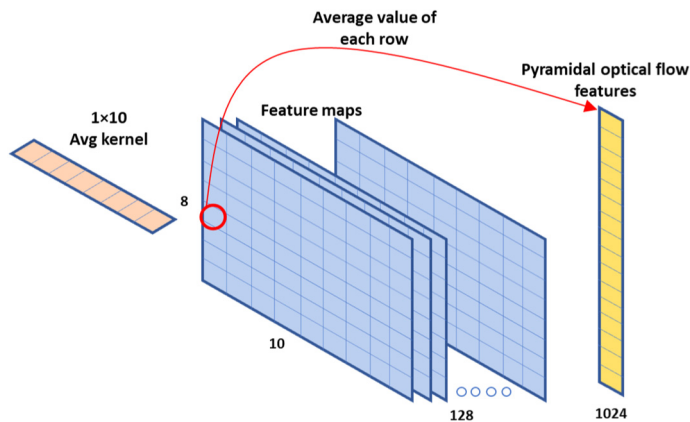


Fig. 3. Procedure of features selection from the convolutional layer of the LiteFlowNet. The feature maps dimensions are  $8 \times 10 \times 128$ , and 1024 is the size of output features vector.

for pedestrian sequence capturing. It is ultra-fast because it can process more than 700 frames per second, and it is robust to illumination changes, different scales, poses, and abrupt movements. The results of tracking a pedestrian using the MOSSE filter are shown in Fig. 2, where the pedestrian is tracked without any interruption caused by different poses, even when half of his body is hidden behind the door.

### 3.3. Pyramidal flow features extraction

Human activity is the collection of visual and motion patterns changing in a sequence of video frames [33]. The representation of visual patterns in a still image has achieved state-of-the-art accuracy since AlexNet was introduced. However, visual contents representation in continuous video frames is very challenging. For this purpose, researchers have introduced space volume features [34], spatio-temporal features [35], 3D CNNs [36], etc., however, they still have not achieved convincing results. Along with visual content, motion in the video sequence is also an important feature that can very accurately describe the performed activity. The descriptor most commonly used to capture motion information is optical flow; however, capturing tiny movements and extracting precise flow features in a sequence of frames in real-time is a difficult task. To deal with these challenges, we exploited a CNN-assisted optical flow model called LiteFlowNet [37]. It is a modified version of FlowNet2 that is faster and more capable of capturing slight motion displacements, which helps to represent real-world human activities in video data. Thus, we claim that the pyramidal convolutional features of LiteFlowNet can efficiently represent activities in video data due to the extraction of large and small displacements in consecutive frames.

The learned features in the intermediate layers of the CNN model are very powerful, and they can be utilized for tasks other than they are originally trained for. Generally, the early layers capture local descriptors, and the final layers of the CNN model represent the high-level semantics of visual data [38].

Therefore, the local information of LiteFlowNet can be utilized to capture local motion patches for activity recognition. LiteFlowNet is an end-to-end deep architecture that is trained using supervised learning on the labeled optical flow of two images to generate flow between two consecutive frames. It contains two networks, NetC and NetE. NetC applies convolutional filters to extract feature descriptors, while NetE uses deconvolutional filters to construct the optical flow between the pair of images. The whole processing pipeline of LiteFlowNet is similar to FlowNet2. First, it processes the pair of images separately to extract the semantic features of each image using NetC, and then the correlation layer combines the semantic representations of both images to exhibit the change between them.

In the proposed technique, we have utilized the middle layer (conv1\_D1\_L6) of the LiteFlowNet model for pyramidal feature extraction. We argue that this layer is capable of the extraction of motion present in the pair of images because it is right after the correlation layer, which performs multiplicative patch comparisons between two feature maps that are taken from the pyramidal pipeline in NetC. The feature maps of this convolutional layer contain 128 channels with dimensions of  $10 \times 8$ . We convolve an average pooling of 101 to each feature map and get only 8 representative features from it. The process of the pyramidal feature selection from the convolutional layer is presented in Fig. 3, where the extracted  $128 \times 10 \times 8$  feature maps are convolved with a  $10 \times 1$  average kernel. Each feature map outputs 8 representative features, which finally become  $1024 \times 128 \times 8$ . Average pooling is used because it has been proven in recent studies that it can very precisely reduce the dimensionality by capturing the effect of all features in the kernel as their mean value. The size of output features for the image pair is 1024. These features are fed to our trained DS-GRU at one-time step, and for a one-second video sequence we pass 15 time-steps, which give us the prediction of the activity and its confidence score.

### 3.4. Learning activity patterns via DS-GRU

Recurrent neural networks (RNN) are an extension of the traditional feedforward neural networks that are specifically designed for sequence learning tasks. RNNs can process time series data by utilizing recurrent hidden states, whose activation at each time step in a sequence is dependent on the activation value of the previous time step(s) [39]. However, it has been observed in many studies that the standard RNN runs into the vanishing gradient problem when the sequences have long term dependencies [22]. To tackle this issue, two dominant variants of RNNs have been introduced by researchers, i.e., LSTM [23] and GRU [40]. The LSTM contains gated recurrent units, including the input, forget, and output gates and a memory cell, while the GRU consists of the reset gate, the update gate, and an activation unit. The structure of the LSTM is more complex and contains gates and a memory cell, resulting in more computations needed to process a sequence. On the other hand, the GRU involves only two gates, which make it applicable for processing real-time sequential data. However, many researchers have reported that the LSTM is more effective than the GRU [40]. In this paper, instead of employing fully auto connected GRU layers, we proposed a DS-GRU network

that employs additional auxiliary connections among GRU units. Generally, the deeper structure of LSTM or GRU models creates the problem of the vanishing gradient, which is solved in this framework by introducing skip connections among GRU layers. Initially, the famous ResNet CNN [29] model utilized skip connections between convolutional layers, which we inflated to the GRU's network to solve the degradation problem and make a possible deeper design for precise accuracy. We link the lower layer output to the higher layer input, and this shortcut connection has been verified to overcome the problem of the vanishing gradient during training and attain expressively better performance. The research in [41] visualized the internal structure of different deep learning models and proved that the skip connection concept also helps to keep the loss function from being chaotic, which leads to a more convex loss and makes it easy to find the local minimum. This architecture makes our system able to achieve LSTM-level accuracy while being more efficient than the LSTM.

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j \check{h}_t^j \quad (1)$$

$$z_t^j = \sigma(W_z x_t + U_z h_t^j) \quad (2)$$

$$\check{h}_t^j = \tanh(Wx_t + U(r_t \circ h_{t-1}))^j \quad (3)$$

$$r_t^j = \sigma(W_r x_t + U_r h_t^j) \quad (4)$$

The GRU adaptively represents the sequential dependencies in time series data. It has recurrent units like LSTM that propagate the flow of patterns in recurrent units, but it has no isolated memory cell for sequence capturing. The processing of the GRU is mathematically given in Eq. (1) to Eq. (4). The  $h_t^j$  activation of each  $j$ th GRU unit at time step  $t$  is calculated using the linear interpolation of  $h_{t-1}^j$ , the hidden state at the previous time step, and  $\check{h}_t^j$ , the activation of the current hidden state. In the proposed case,  $h_t^j = 1024$  features are extracted from two consecutive video frames at time  $t$ . The parameter  $z_t^j$  is the update gate and works as a special kernel that decides how much the activation of each unit needs to be updated. Some steps of the GRU are similar to the LSTM, such as the linear sum between the previous hidden state and the current state. The parameter  $\check{h}_t^j$  is the hidden state at the current time step that is computed as in traditional RNNs. The parameter  $r_t$  is the reset gate, and when it is close to 0, it effectively forgets the information of the previously computed state.

The deep hierarchical way of learning hidden patterns in complex real-life visual data has proven itself useful in various fields, such as image retrieval [42], object detection and localization [43], video summarization [44,45], etc. The proposed technique implies the strategy of stacking multiple GRUs with skip connections concept to effectively learn long-term sequential patterns. The hidden layer at the current time step in the GRU takes data from the previous hidden state. Similarly, in stacked GRUs, the input to the upper layer GRU is the data from the previous layers of GRUs. However, in our skip connections network, the output of lower layer GRU is concatenated with the higher layer GRU in the network. Let us suppose,  $O_1 = \text{GRU}_1(x)$ , where  $O_1$  is the output of a first GRU layer, then  $O_2 = \text{GRU}_2(O_1)$ ,  $O_3 = \text{GRU}_3(O_2)$ ,  $O_4 = \text{GRU}_4(O_1 + O_3)$ , and  $O_5 = \text{GRU}_5(O_2 + O_4)$ .

### 3.5. Hyperparameter tuning

We stacked five GRU layers that have two skip connections, as shown in Fig. 1. This network is more efficient than the recently proposed directly auto connected multi-layer LSTM [1]. To train the DS-GRU, first, 15350 dimensional pyramidal features are extracted from a video clip of one second. Next, 1024 features of

**Table 2**

The overall accuracy and performance achieved using of the proposed DS-GRU with different number of layers and hidden states.

Experiments	Hidden states	Number of parameters (millions)	Time complexity for 30 frames (seconds)	Overall accuracy (%)
5 GRU Layers	256	2.5	0.205	95.5
5 GRU Layers	512	8.5	0.531	96.3
7 GRU Layers	256	3.3	0.421	96.0
7 GRU Layers	512	11.8	0.958	96.3

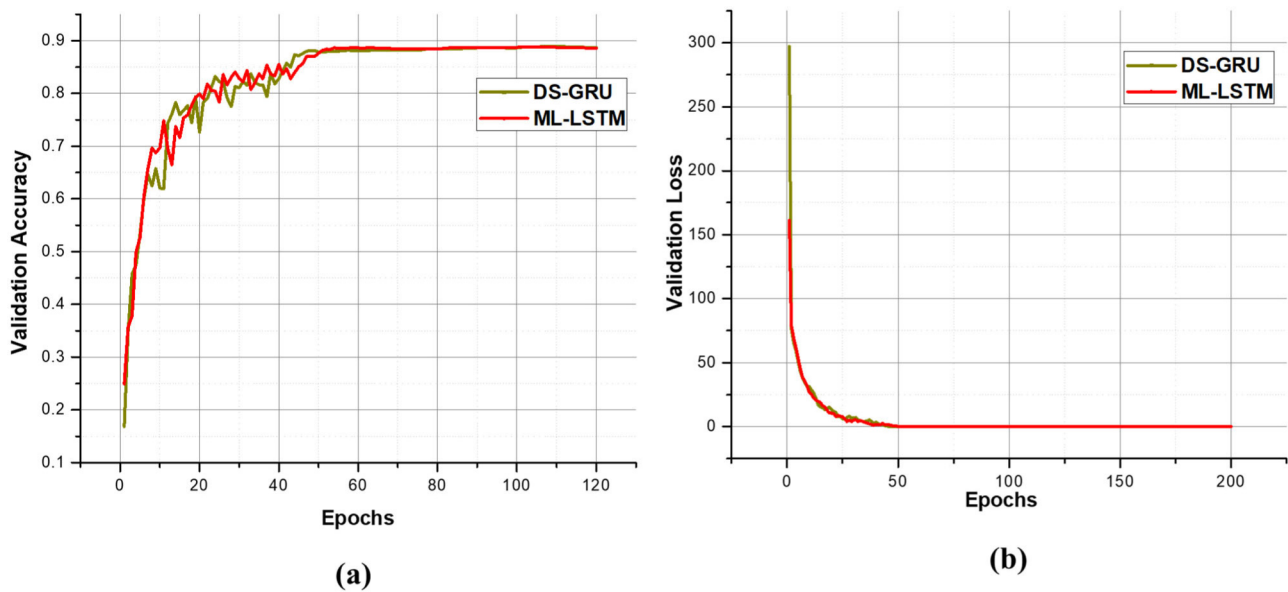
two consecutive frames are input to the DS-GRU at a particular time step. The selections of the number of layers and the trainable parameters are very important in any neural network model [46]. We conducted various experiments and finally selected 5 layers and 256 hidden states in each GRU. From our experiments, we concluded that adding more layers and trainable parameters increases the inference model size; however, its performance is the same as it is with 5 layers. The ablation study of layers and parameters selection is given in Table 2. For instance, we mentioned that each GRU of the network consists of 256 hidden states. We also tested the network with 512 and 1024 hidden states, but the results are same as for 256 hidden states. The proposed model is trained for 200 epochs to learn sequential patterns, where stochastic optimization is used for cost minimization, dropout is set to 0.5 to avoid the overfitting problem, and the initial learning rate is 0.01, which is decreased by a factor of 10 after 50 epochs. Fig. 4 demonstrates the performance of the DS-GRU and multi-layer LSTM on the YouTube Actions dataset. The validation loss can be observed from Fig. 4(b), which indicates that the proposed DS-GRU can learn complex hidden sequential patterns effectively without falling into the overfitting problem.

## 4. Experimental results and discussion

In this section, our proposed activity recognition technique is evaluated in experiments over various benchmark datasets. The metrics used for evaluation are the overall accuracy, class-wise accuracy, confusion matrix, receiver operating characteristic curve (ROC), area under the curve (AUC) values, training model size, and time complexity. For these metrics, our technique is extensively compared with existing approaches for activity recognition. The experimental environment consists of Python 2.7 installed on Ubuntu-16.04, a setup with a Core™i5-6600 processor with 16 GB RAM and supplied with the support of a dedicated 12 GB GeForce-Titan-X GPU. The 'Caffe' and 'Tensorflow' deep learning toolboxes are utilized for pyramidal flow feature extraction and implementation of the DS-GRU, respectively.

### 4.1. Benchmark datasets used in our experiments

The experiments have been conducted using the most challenging datasets, including HMDB51 [58], UCF-101 [59], UCF-50 [60], Hollywood2 Actions [61], and YouTube Actions [62]. Each dataset comprises multiple videos having different durations, where the HMDB51 dataset [58] consists of 51 distinct activity categories and is a collection of a total of 6474 video clips from 1697 unique sources. In the HMDB51 dataset, each clip in every activity category is annotated with a label and meta-label to describe the clip's properties, which can include camera motion, viewpoint, and number of involved people in the actions. The UCF-101 [59] is an extension of UCF-50 [60], which contains 50 activity classes, including basketball, clean and jerk, drumming, skydiving, and so on. UCF-101 is a large-scale dataset consisting of 101 activity categories and 13320 videos in total. This dataset provides a diverse array of activities, with the presence



**Fig. 4.** Sequential patterns learning of the proposed DS-GRU and multi-layer LSTM for 200 training epochs: (a) Validation accuracy and (b) Validation loss. Our proposed DS-GRU has achieved LSTM-level performance on the YouTube Actions dataset, while being 1.4 times faster than the multi-layer LSTM network.

---

#### Algorithm 1. Activity Recognition using Pyramidal Flow Features and a DS-GRU Network

---

**Input:** Continuous video frames

**Output:** Predicted activity class with confidence score

**Preparation:**

1. YOLOv3 pedestrian detection model
2. MOSSE visual object tracker
3. LiteFlowNet CNN model
4. Trained DS-GRU

**Steps:**

1. **While** (video frame)
2. ROIs  $\leftarrow$  Feed frame to trained YOLOv3 CNN
3. **If** (ROIs)
  - a. MOSSE tracker  $\leftarrow$  ROIs
  - b. Propagate forward tracked ROI<sub>i</sub> and ROI<sub>i+1</sub> of pedestrian sequence to LiteFlowNet CNN
  - c. Extract 128×10×8 feature maps from layer ‘conv1\_D1\_L6’
  - d. Apply horizontal average pooling 10×1 to each row of feature map to get pyramidal feature vector  $F_v$
  - e.  $F_v$  is fed to DS-GRU at step ‘t’
4. Labeled Predict activity  $\leftarrow$  Trained DS-GRU
5. Display predicted activity with confidence score

**End If**

**End While**

---

of different camera motion variations, object scale variations, and cluttered backgrounds. According to the original paper [59], this dataset is divided into five categories: sports, playing musical instruments, human-to-human interaction, human-to-object interaction, and body motion only. Besides these datasets, we included Hollywood2 [61] and YouTube Actions [62], consisting of 16 and 11 classes, respectively. The Hollywood dataset consists of 937 clips having 787720 frames, which contain sequences taken from 69 Hollywood movies. The YouTube Actions dataset consists of 1160 videos, including such activities as diving, tennis, football, golf, and horse riding. This dataset is created with 25 different subjects, where each subject has provided more than four clips. In these datasets, the large variations are due to camera motion, pose appearance, and illumination conditions, which made these datasets more inspiring and challenging.

#### 4.2. Assessment with the state-of-the-art techniques

The proposed technique is compared with deep LSTM and non-LSTM-based techniques, as demonstrated in Table 3. The overall accuracies are given in Table 3, where the highest accuracy is represented in bold and the runner-up is underlined. Our technique has the highest accuracy on three datasets and achieved the runner-up position on two datasets. The video LSTM [22] has the uppermost accuracy of 73.3% and the proposed technique achieved the second highest accuracy of 72.34% on the HMDB51 dataset. On this dataset, the bi-directional LSTM [18], relational LSTM [49], temporal optical flow with multi-layer LSTM [1], 3D CNNs, and hierarchical LSTM [51] achieved 70.4%, 71.4%, 72.21%, and 71.9% accuracies, respectively. The other methods have accuracies less than 70%. On the UCF-101 dataset, the relational

**Table 3**

Comparison of our proposed DS-GRU with LSTM and non-LSTM based activity recognition methods for five benchmarked datasets. The highest accuracy is represented in bold and the runner-up is underlined. The bold and italic text represent first and second highest accuracy, respectively.

Method	HMDB51 (%)	UCF-101 (%)	UCF-50 (%)	Hollywood2 (%)	YouTube Actions (%)
ARCH [47]	58.2	85.3	-	63.1	-
Lattice-LSTM [48]	66.2	93.6	-	-	-
Bi-directional LSTM [18]	70.4	94.2	-	-	-
Video LSTM[22]	<b>73.3</b>	92.2	-	-	-
Deep bi-directional LSTM [23]	-	91.2	-	-	92.84
LSTM based Methods					
Relational LSTM [49]	71.4	<u>94.8</u>	-	-	-
TS-LSTM and temporal-inception [50]	69.0	91.1	-	-	-
Temporal optical flow with multi-layer LSTM [1]	72.2	94.4	<u>94.9</u>	<u>69.5</u>	95.8
3D-CNNs and bi-directional hierarchical LSTM [51]	71.9	<u>94.8</u>	-	-	-
Two-stream attention LSTM [52]	-	-	-	-	96.9
Non-LSTM based Methods					
Improved trajectory [4]	57.2	-	91.2	64.3	-
Multi-skip feature stacking (MSFS) [53]	-	-	-	68.0	-
Single stream CNN [54]	-	-	-	-	93.1
Improved dense trajectories [55]	61.1	87.9	92.3	-	-
Hierarchical clustering multi-task [56]	51.4	76.3	93.2	-	89.7
Fusion of handcrafted and CNN [57]	-	-	-	-	<b>100</b>
<b>Proposed DS-GRU</b>	<u>72.3</u>	<b>95.5</b>	<b>95.2</b>	<b>71.3</b>	<u>97.17</u>

**Table 4**

Evaluation of the proposed technique using precision, recall, and F1-score.

Dataset	Precision (%)	Recall (%)	F1-Score (%)
UCF-101	86.398	83.542	82.120
UCF-50	91.294	89.647	88.637
HMDB51	64.982	61.952	60.734
Hollywood2	68.219	66.846	65.797
YouTube Actions	92.637	91.541	91.436

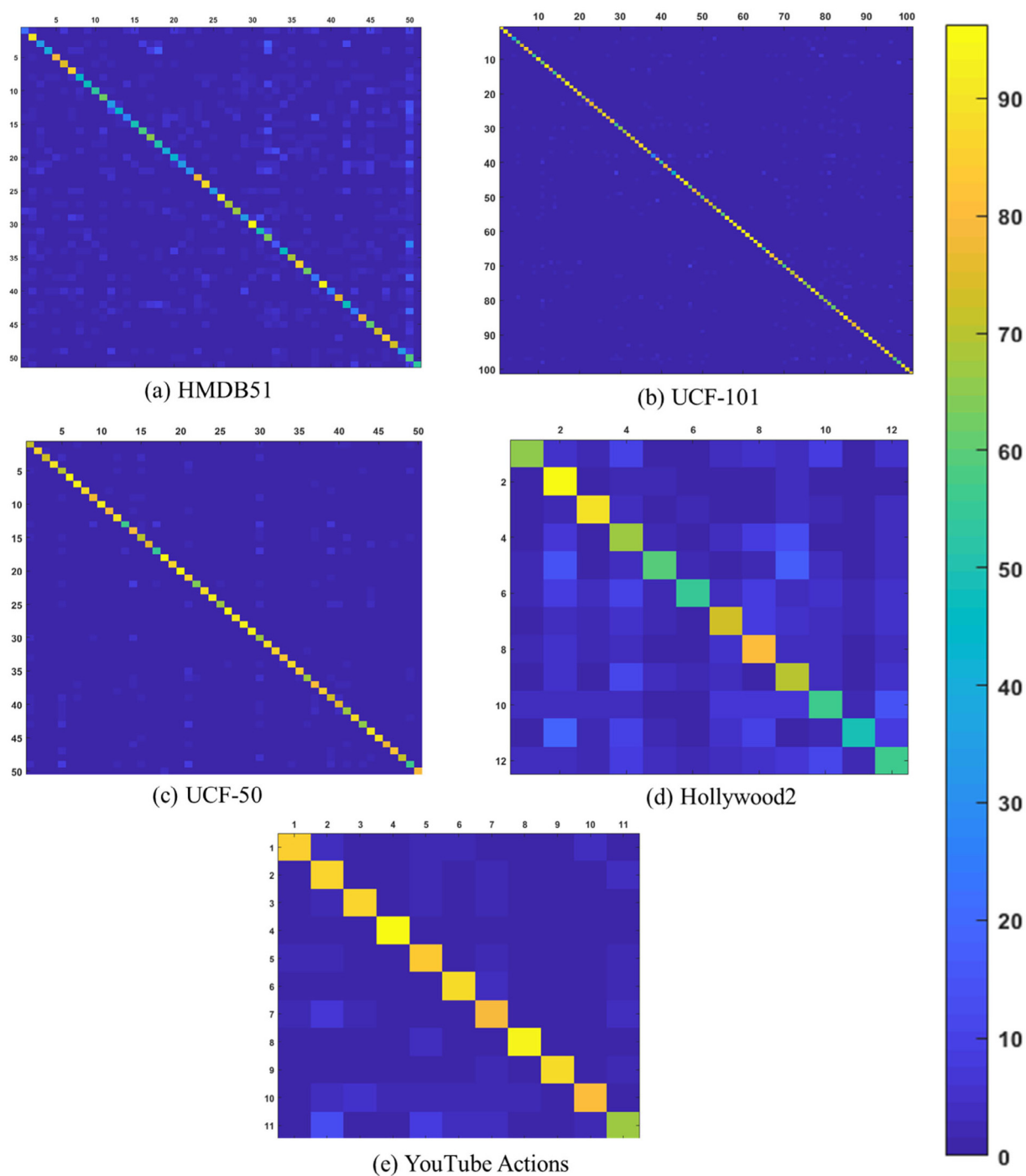
LSTM [49], 3D CNNs, and bi-directional hierarchical LSTM [51] achieved 94.8% accuracies, and the proposed technique achieved 95.5% accuracy, reaching the top position in Table 3. On the UCF-101 dataset, most of the methods achieved an overall accuracy of approximately 90%. The proposed technique has improved the highest accuracy on the UCF-50 dataset from 94.9% to 95.21%. The improved [4], hierarchical clustering multi-task [56], and improved dense trajectories [55] reached 91.2%, 93.2%, and 92.3% accuracies, respectively. The Hollywood2 dataset contains challenging movie data on which our proposed technique improved the highest accuracy from 69.5% to 71.35%. However, ARCH [47], improved trajectory [4], and MSFS [53] achieved accuracies of 63.1%, 64.3%, and 68.0%, respectively. YouTube Actions is a small dataset of 11 categories, and our proposed technique achieved the highest accuracy of 96.17% on this dataset. The recent fusion of handcrafted and CNN [57] based methods claimed 100% accuracy on this dataset; however, their method performs well on datasets with fewer categories and is not applicable for large-scale datasets. The accuracies of other methods on this dataset are also greater than 90%. The proposed technique has improved the overall achievable accuracy on this dataset and achieved effective results by using lower computational complexity.

The competence of our technique is also assessed using the ROC curve and AUC values, which are visualized in Fig. 6. The ROC calculates the contrast between the true positive rate (TPR) and the false positive rate (FPR) at different threshold values for classification decisions. It can be seen from Fig. 6 that the proposed technique achieved the best ROC curves and AUC values out of all of the datasets. For instance, for the HMDB51 dataset, it achieved 0.956 AUC, and for the UCF-101, UCF-50, Hollywood2 Actions, and YouTube Actions datasets, it achieved AUC values of 0.971, 0.966, 0.944, and 0.988, respectively.

#### 4.3. Class-wise performance, precision, recall, and F1-scores

Our proposed technique is assessed using five accuracy evaluation metrics. The confusion matrices for the test sets of five state-of-the-art datasets are shown in Fig. 5. To assess the positive prediction value and sensitivity of our technique, we have calculated the precision, recall, and F1-measure scores for each dataset, which are given in Table 4. The proposed technique has achieved balanced precision and recall scores for all datasets, indicating less number of true negatives and false negatives. Our technique has achieved F1-scores of 82.12%, 88.63%, 60.73%, 65.79%, and 91.43% for the UCF-101, UCF-50, HMDB51, Hollywood2, and YouTube Actions datasets, respectively, showing the effectiveness of the proposed technique compared to state-of-the-art techniques. The class-wise performance for each dataset is visualized in Fig. 7. For each dataset, the horizontal axis represents categories, and the vertical axis shows percentage accuracies. It can be seen from the graphs that our technique performs well in all categories. On challenging datasets, such as HMDB51 and Hollywood2, the proposed technique achieved score of less than 50% for some categories; however, for UCF-101, UCF-50 and YouTube Actions, we achieved more than 80% average class accuracy.





**Fig. 5.** Confusion matrices achieved using the proposed DS-GRU for the test sets of five benchmark datasets, including (a) HMDB51, (b) UCF-101, (c) UCF-50, (d) Hollywood2, and (e) YouTube Actions.

#### 4.4. Network parameters and computational complexity analysis

The running time analysis and the model size of our proposed technique are examined in this section. The time complexity analysis and comparison are visualized in Fig. 8. Fig. 8 (a) shows the time required for all subsequent tasks for the processing of one video sequence, while Fig. 8(b) shows the comparison of our method with state-of-the-art activity recognition methods. Human detection takes 85 ms, tracking the detected person for one second using MOSSE algorithm takes 103 ms, feature extraction takes 630 ms, and activity classification requires 205 ms. The experiments are performed only on a GPU setup because the

model we utilized has no CPU implementation version since it has some special convolutional layers. The ML-LSTM [1], which utilized FlowNet2 features, takes 1.068 s to process a 1-second video on a GPU, and it takes 5.4 s on a CPU. The deep autoencoder with QSVM [63] needs 1.31 s to process 30 frames on a GPU and 13 s on a CPU. The proposed technique is more efficient and can process 30 FPS video data in 0.83 s. A comparison of model sizes is given in Table 5. The well-known C3D model has a large size of 321 MB due to its usage of large 3D filters. The MiCT-Net utilized 3D and 2D filters combined to reduce the model size to 221 MB. The proposed technique has fewer parameters and small size because we utilized a GRU, which does not have a memory



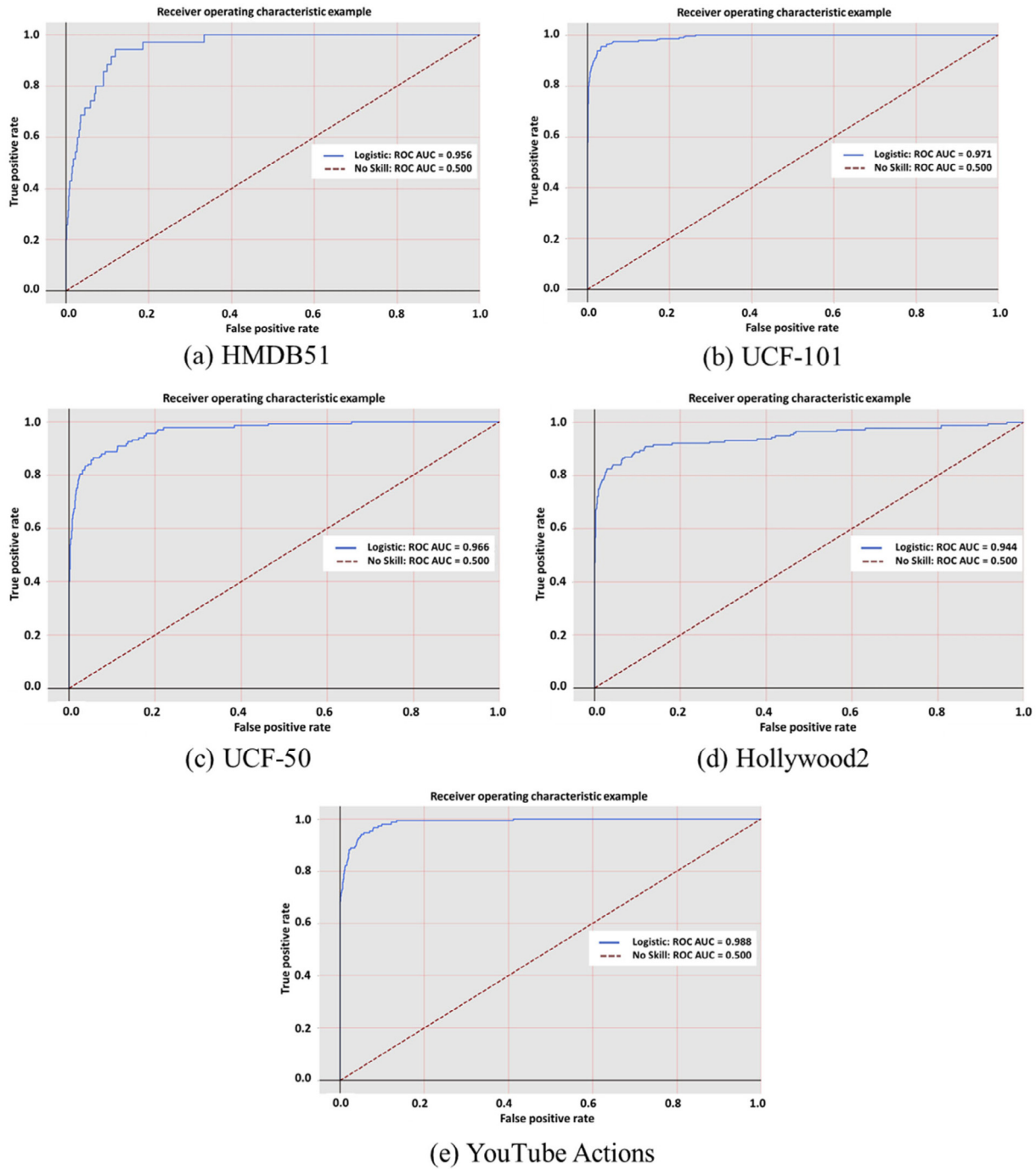


Fig. 6. ROC curve and AUC values achieved using the proposed DS-GRU for the test sets of five benchmark datasets, including (a) HMDB51, (b) UCF-101, (c) UCF-50, (d) Hollywood2, and (e) YouTube Actions.

cell like the LSTM. This makes it suitable for implementation in real-time surveillance for effective activity recognition.

#### 4.5. Tracking results and discussion

Tracking is an important part of the proposed framework because a real-time correctly tracked area results in precise activity recognition results. For this purpose, we investigated various tracking techniques and utilized the MOSSE tracker for activity recognition due to its fast and accurate results. Visual results comparisons of various tracking algorithms are given in Fig. 9, along with their processing rates in FPS. The state-of-the-art methods including TLD [65], Boosting [66], CSRT [67], and

KCF [68] drift in fast motion and challenging scenarios. Additionally, their processing rates are very slow. On the other hand, it can be seen from Fig. 9 that MOSSE has achieved better results with a very high FPS, which helps in real-time activity recognition. It is true that the MOSSE tracker is prone to drifting. However, most of the drifting and failures occur when the tracking target undergoes a large out-of-plane rotation. This problem primarily occurs when Naïve filtering is used in MOSSE [69], and in the proposed framework a Peak-to-Sidelobe Ratio filtering (MOSSE PSR) instead of Naïve filters. MOSSE PSR filtering measures the strength of a correlation peak, which can be used to detect occlusions or tracking failure, to stop the online update and to reacquire the track if the object reappears with a similar appearance.

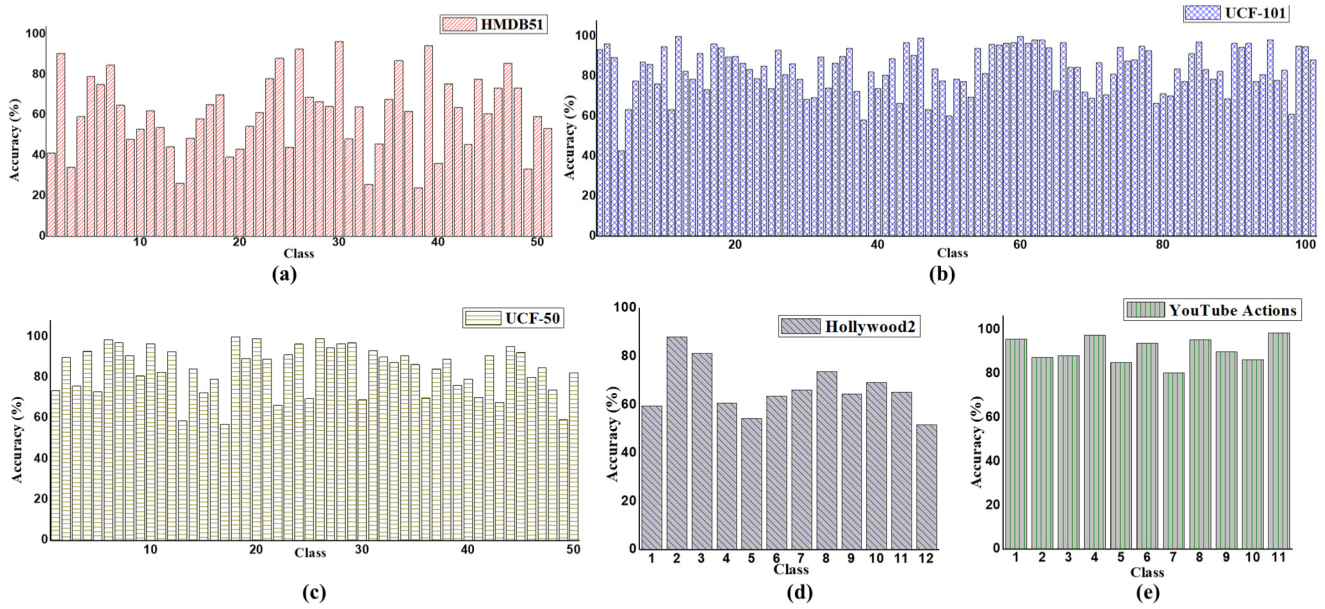


Fig. 7. Class-wise performance of our technique for the test sets of five benchmarked datasets, including (a) HMDB51, (b) UCF-101, (c) UCF-50, (d) Hollywood2, and (e) YouTube Actions.

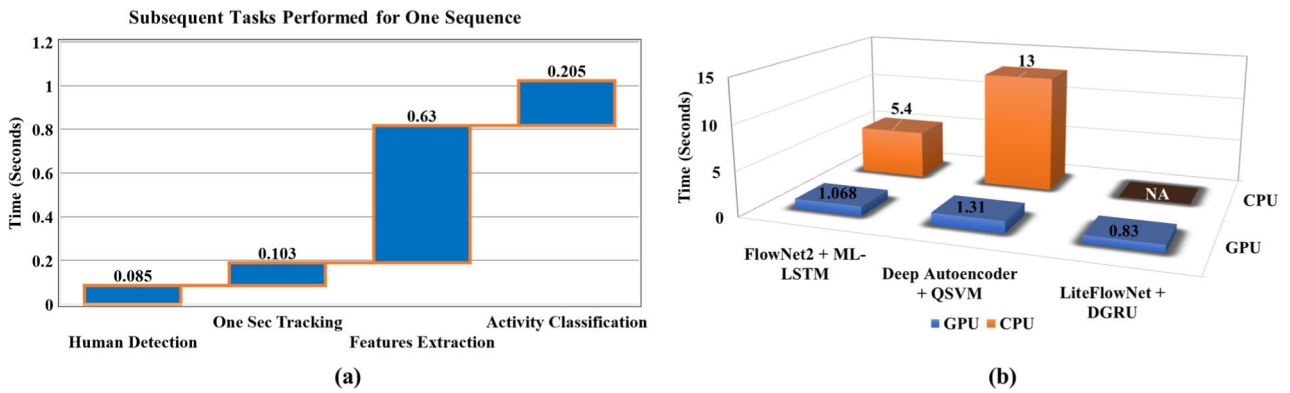


Fig. 8. Time complexity analysis of the proposed technique: (a) Time required for all subsequent tasks for one video sequence analysis and (b) Comparison with other state-of-the-art methods for only the activity recognition step.

Table 5

A comparison of the proposed technique with other deep learning-based techniques based on the number of weighted layers, parameters, and trained model size.

Model name	Number of weighted layers	Number of parameters (millions)	Model size (MB)
C3D [64]	11	~	321
MiCT-Net [10]	~	~	221
FlowNet2 + ML-LSTM [1]	115 + 4 = 119	162.49 + 2.36 = 164.85	163 + 30 = 193
<b>LiteFlowNet + DS-GRU</b>	99 + 4 = 103	5.37 + 2.55 = 7.29	28 + 22 = 48

Table 6

Performance of state-of-the-art deep learning-based tracking models.

Year	Method	Backend CNN model	FPS	Model layers
2018	MOTDT [70]	R-FCN	20.6	10
2018	COMOT [71]	Part-based deep network	5.16	15
2019	GM-PHD-DAL [72]	ResNet50	3.5	50
2020	OneShotDA [73]	Feature embedding network and conditional embedding network	3.5	13
2020	Siam R-CNN [74]	Faster R-CNN	15.2	15
2020	HMB_DAF [75]	CNN with Residual Blocks	37.6	15











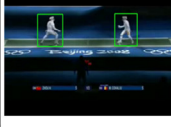
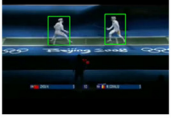

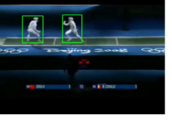

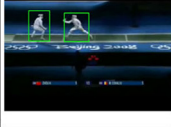
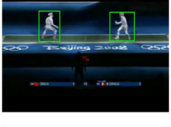
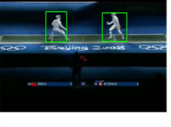

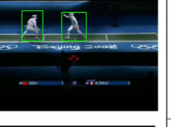
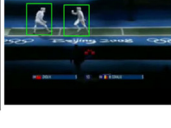




Method	Intermediate frames tracking visual results					Processing (FPS)
TLD						22
Boosting						15
CSRT						28
KCF						44
MOSSE						275

Fig. 9. Visual comparison of the results achieved using various tracking algorithms.

Furthermore, it is useful when tracking a very small area inside a sequence of frames. However, we are tracking the contents of a full pedestrian body, which is large enough to stop drifting in tracking. On the other hand, there exist several deep learning-based trackers that have outperformed traditional trackers. However, implementation of these deep trackers requires dedicated GPU usage [76]. Moreover, using a GPU, the competence and performance of deep trackers are very good, but executing these trackers on a CPU reduces the 30 FPS performance to a very low frame rate [77]. On the other hand, in the proposed framework, two deep learning models (LiteFlowNet for features extraction and a DS-GRU) are already used, which are the actual tasks for the framework, while the tracking is only used as a preprocessing step to analyze the actual area of the human activity. The MOSSE tracker has adequate performance on a CPU, therefore, we used it in the proposed framework. It should be kept in mind that the tracking is performed using a CPU, and the activity recognition task is performed on a GPU. Table 6 shows the performance of deep learning-based tracking algorithms, where the frames per second (FPS) column clearly states that if we use any of these models, it requires dedicated GPU and can track objects in less than 30 frames in one second. However, alongside tracking, two other tasks are also being executed over the GPU. Similarly, these trackers are using very deep CNN models with huge number of layers. Therefore, with GPU having only 12 GB of memory, we cannot load three deep learning models together. Therefore, we have used MOSSE tracking algorithm over the CPU and the rest of two tasks over the GPU.

## 5. Conclusions and future work

Human activity recognition has been a hot area of research in the last two decades, and several low-level and high-level features-based techniques integrated with different classifiers are proposed. Mainstream state-of-the-art techniques proposed for

activity recognition are based on heavyweight CNN models that aim to get better accuracy. It is hard to maintain a good trade-off between accuracy and efficiency in activity recognition systems reported in the literature, due to the processing of high dimensional video data. Precise activity classification models are computationally complex, making quick responsive actions inflexible if there are abnormal activities. To target these challenges efficiently and accurately, we proposed a real-time technique for activity recognition. It comprises several steps, including human detection, tracking, learned features extraction, and the use of a DS-GRU for activity classification. Human detection is a vital step in activity recognition, therefore, we utilized a lightweight CNN model to detect humans. Pyramidal convolutional features from each tracked pedestrian are extracted using a fast LiteFlowNet CNN model for two consecutive frames. Finally, the activity is classified using a DS-GRU that is trained to learn the temporal changes in a sequence of frames. Extensive experiments over different activity recognition datasets confirm the effectiveness of the approach, and the time complexity analysis proves the efficiency of our proposed framework.

Currently, our proposed framework has some drawbacks that we wish to target in future research. In this paper, we aimed at single-view activity recognition that cannot provide full 360-degree coverage of an activity. In future research, we will analyze multi-view data for signal and group activity recognition [78,79] using the collaboration of multiple sensors. Furthermore, as the current system run efficiently using the GPU, we will transform our framework to embedded platforms to perform activity recognition over the edge. The current methodology can be optimized for resource constrained devices such as nano-Jetson, google board, and R-Pi. The final thing we want to discuss is possible ethical concerns related to the proposed method and its positive and negative impact on society, which is very important for any artificial intelligence framework [80]. The proposed method can help law enforcement agencies to recognize abnormal activities in surveillance such as accidents, activities that



damage property or are against the law [81], criminal activities such as fights, stealing, etc. However, it may generate false alarms because the current version is trained on publicly available data of different activities, and it will be hard to directly implement it in surveillance settings. This will require proper activities data collected from many surveillance settings and fine-tuning of the method we proposed for actual implementation.

### CRedit authorship contribution statement

**Amin Ullah:** Conceptualization, Writing - original draft. **Khan Muhammad:** Data curation, Methodology. **Weiping Ding:** Writing - review & editing, Investigation. **Vasile Palade:** Writing - review & editing. **Ijaz Ul Haq:** Validation, Visualization. **Sung Wook Baik:** Supervision.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgment

This work was supported by the National Research Foundation of Korea and funded by the Korean government (MSIP) under Grant No. 2019R1A2B5B01070067.

### References

- [1] A. Ullah, K. Muhammad, J.D. Ser, S.W. Baik, V.H.C.d. Albuquerque, Activity recognition using temporal optical flow convolutional features and multilayer LSTM, *IEEE Trans. Ind. Electron.* 66 (12) (2019) 9692–9702.
- [2] B. Yousefi, C.K. Loo, A dual fast and slow feature interaction in biologically inspired visual recognition of human action, *Appl. Soft Comput.* 62 (2018) 57–72.
- [3] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, K. Tang, Kernel fusion based extreme learning machine for cross-location activity recognition, *Inf. Fusion* 37 (2017) 1–9.
- [4] H. Wang, C. Schmid, Action recognition with improved trajectories, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.
- [5] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, J. Huang, End-to-end learning of motion representation for video understanding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6016–6025.
- [6] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.
- [7] Y. Shi, Y. Tian, Y. Wang, T. Huang, Sequential deep trajectory descriptor for action recognition with three-stream CNN, *IEEE Trans. Multimed.* 19 (7) (2017) 1510–1520.
- [8] X. Wang, L. Gao, P. Wang, X. Sun, X. Liu, Two-stream 3-D convnet fusion for action recognition in videos with arbitrary size and length, *IEEE Trans. Multimed.* 20 (3) (2017) 634–644.
- [9] Z. Qiu, T. Yao, T. Mei, Learning spatio-temporal representation with pseudo-3d residual networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.
- [10] Y. Zhou, X. Sun, Z.-J. Zha, W. Zeng, MiCT: Mixed 3D/2D convolutional tube for human action recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 449–458.
- [11] R. Zhao, H. Ali, P. Van der Smagt, Two-stream RNN/CNN for action recognition in 3D videos, in: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 4260–4267.
- [12] M. Majd, R. Safabakhsh, Correlational convolutional LSTM for human action recognition, *Neurocomputing* 396 (2020) 224–229.
- [13] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, IEEE, 2010, pp. 2544–2550.
- [14] A. Dosovitskiy, et al., FlowNet: Learning optical flow with convolutional networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.
- [15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, FlowNet 2.0: Evolution of optical flow estimation with deep networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2462–2470.
- [16] S. Herath, M. Harandi, F. Porikli, Going deeper into action recognition: A survey, *Image Vis. Comput.* 60 (2017) 4–21.
- [17] L.M. Dang, K. Min, H. Wang, M.J. Piran, C.H. Lee, H. Moon, Sensor-based and vision-based human activity recognition: A comprehensive survey, *Pattern Recognit.* (2020) 107561.
- [18] W. Li, W. Nie, Y. Su, Human action recognition based on selected spatio-temporal features via bidirectional LSTM, *IEEE Access* 6 (2018) 44211–44220.
- [19] H. Gammulle, S. Denman, S. Sridharan, C. Fookes, Two stream lstm: A deep fusion framework for human action recognition, in: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 177–186.
- [20] L. Sun, K. Jia, K. Chen, D.-Y. Yeung, B.E. Shi, S. Savarese, Lattice long short-term memory for human action recognition, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2147–2156.
- [21] S. Ma, L. Sigal, S. Sclaroff, Learning activity progression in lstms for activity detection and early detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.
- [22] Z. Li, K. Gavriljuk, E. Gavves, M. Jain, C.G. Snoek, VideoLstm convolves attends and flows for action recognition, *Comput. Vis. Image Underst.* 166 (2018) 41–50.
- [23] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, S.W. Baik, Action recognition in video sequences using deep Bi-directional LSTM with CNN features, *IEEE Access* 6 (2018) 1155–1166.
- [24] H. Kuehne, A. Richard, J. Gall, A hybrid rnn-hmm approach for weakly supervised temporal action segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).
- [25] M. Qi, Y. Wang, J. Qin, A. Li, J. Luo, L. Van Gool, Stagnet: An attentive semantic RNN for group activity and individual action recognition, *IEEE Trans. Circuits Syst. Video Technol.* 30 (2) (2019) 549–565.
- [26] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
- [27] D. Wang, C. Zhang, H. Cheng, Y. Shang, L. Mei, SPID: surveillance pedestrian image dataset and performance evaluation for pedestrian detection, in: *Asian Conference on Computer Vision*, Springer, 2016, pp. 463–477.
- [28] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (4) (2012) 743–761.
- [29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [30] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.
- [31] J. Redmon, A.J.a.p.a. Farhadi, Yolov3: An incremental improvement, 2018.
- [32] X. Dai, B. Singh, G. Zhang, L.S. Davis, Y. Qiu Chen, Temporal context network for activity localization in videos, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5793–5802.
- [33] Y. Shan, Z. Zhang, P. Yang, K. Huang, Adaptive slice representation for human action classification, *IEEE Trans. Circuits Syst. Video Technol.* 25 (10) (2015) 1624–1636.
- [34] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Vol. 1, 2*, IEEE, 2005, pp. 1395–1402.
- [35] K. Hara, H. Kataoka, Y. Satoh, Learning spatio-temporal features with 3D residual networks for action recognition, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3154–3160.
- [36] H. Xu, A. Das, K. Saenko, R-c3d: Region convolutional 3d network for temporal activity detection, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5783–5792.
- [37] T.-W. Hui, X. Tang, C. Change Loy, LiteFlowNet: A lightweight convolutional neural network for optical flow estimation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8981–8989.
- [38] A.M. Badshah, et al., Deep features-based speech emotion recognition for smart affective services, 78, (5) 2019, pp. 5571–5589.
- [39] J. Donahue, et al., Long-term recurrent convolutional networks for visual recognition and description, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.
- [40] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.
- [41] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, in: *Advances in Neural Information Processing Systems*, 2018, pp. 6389–6399.



- [42] I. Mehmood, et al., Efficient image recognition and retrieval on IoT-assisted energy-constrained platforms from big data repositories, *IEEE Internet Things J.* 6 (6) (2019) 9246–9255.
- [43] K. Muhammad, S. Khan, M. Elhoseny, S.H. Ahmed, S.W. Baik, Efficient fire detection for uncertain surveillance environment, *IEEE Trans. Ind. Inform.* 15 (5) (2019) 3113–3122.
- [44] K. Muhammad, T. Hussain, S.W. Baik, Efficient CNN based summarization of surveillance videos for resource-constrained devices, *Pattern Recognit. Lett.* (2018).
- [45] Z. Gao, et al., Salient object detection in the distributed cloud-edge intelligent network, *IEEE Netw.* (2020) 1–9.
- [46] L. Oneto, *Model Selection and Error Estimation in a Nutshell*, Springer, 2020.
- [47] M. Xin, H. Zhang, H. Wang, M. Sun, D. Yuan, Arch: Adaptive recurrent-convolutional hybrid networks for long-term action recognition, *Neurocomputing* 178 (2016) 87–102.
- [48] L. Sun, K. Jia, K. Chen, D.-Y. Yeung, B.E. Shi, S. Savarese, Lattice long short-term memory for human action recognition, in: *ICCV*, 2017, pp. 2166–2175.
- [49] Z. Chen, B. Ramachandra, T. Wu, R.R. Vatsavai, Relational long short-term memory for video action recognition, 2018, arXiv preprint arXiv:1811.07059.
- [50] C.-Y. Ma, M.-H. Chen, Z. Kira, G. AlRegib, TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition, *Signal Process., Image Commun.* 71 (2019) 76–87.
- [51] H. Yang, J. Zhang, S. Li, T. Luo, Bi-direction hierarchical LSTM with spatial-temporal attention for action recognition, *J. Intell. Fuzzy Systems*, no. Preprint, pp. 1–12.
- [52] C. Dai, X. Liu, J. Lai, Human action recognition using two-stream attention based LSTM networks, *Appl. Soft Comput.* 86 (2020) 105820.
- [53] Z. Lan, M. Lin, X. Li, A.G. Hauptmann, B. Raj, Beyond gaussian pyramid: Multi-skip feature stacking for action recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 204–212.
- [54] S. Ramasinghe, R. Rodrigo, Action recognition by single stream convolutional neural networks: An approach using combined motion and static information, in: *Pattern Recognition (ACPR)*, 2015 3rd IAPR Asian Conference on, IEEE, 2015, pp. 101–105.
- [55] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice, *Comput. Vis. Image Underst.* 150 (2016) 109–125.
- [56] A.-A. Liu, Y.-T. Su, W.-Z. Nie, M. Kankanhalli, Hierarchical clustering multi-task learning for joint human action grouping and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (1) (2017) 102–114.
- [57] M.A. Khan, M. Sharif, T. Akram, M. Raza, T. Saba, A. Rehman, Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition, *Appl. Soft Comput.* 87 (2020) 105986.
- [58] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: *Computer Vision (ICCV)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 2556–2563.
- [59] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, 2012, arXiv preprint arXiv:1212.0402.
- [60] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, *Mach. Vis. Appl.* 24 (5) (2013) 971–981.
- [61] M. Marszałek, I. Laptev, C. Schmid, Actions in context, in *computer vision and pattern recognition*, in: 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2929–2936.
- [62] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1996–2003.
- [63] A. Ullah, K. Muhammad, I.U. Haq, S.W. Baik, Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments, *Future Gener. Comput. Syst.* 96 (2019) 386–397.
- [64] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.
- [65] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (7) (2012) 1409–1422.
- [66] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: *European Conference on Computer Vision*, Springer, 2008, pp. 234–247.
- [67] A. Lukežič, T. Vojir, L. Čehovin Zajc, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6309–6318.
- [68] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3) (2015) 583–596.
- [69] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 2544–2550.
- [70] L. Chen, H. Ai, Z. Zhuang, C. Shang, Real-time multiple people tracking with deeply learned candidate selection and person re-identification, in: *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1–6.
- [71] C. Xu, Y. Zhou, Consistent online multi-object tracking with part-based deep network, in: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, Springer, 2018, pp. 180–192.
- [72] N.L. Baisa, Online multi-object visual tracking using a GM-PHD filter with deep appearance learning, in: *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, 2019, pp. 1–8.
- [73] K. Yoon, J. Gwak, Y.-M. Song, Y.-C. Yoon, M.-G. Jeon, Oneshotda: Online multi-object tracker with one-shot-learning-based data association, *IEEE Access* 8 (2020) 38060–38072.
- [74] P. Voigtlaender, J. Luiten, P.H. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6578–6588.
- [75] Q. Ji, H. Yu, X. Wu, Hierarchical-matching-based online and real-time multi-object tracking with deep appearance features, *Algorithms* 13 (4) (2020) 80.
- [76] S.J.a.p.a. Murray, Real-time multiple object tracking—a study on the importance of speed, 2017.
- [77] S. Hossain, D.-j.j.S. Lee, Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices, 19, (15) 2019, p. 3371.
- [78] R. Yan, J. Tang, X. Shu, Z. Li, Q. Tian, Participation-contributed temporal dynamic model for group activity recognition, in: *2018 ACM Multimedia Conference on Multimedia Conference*, ACM, 2018, pp. 1292–1300.
- [79] Z. Yan, J. Liu, L.T. Yang, W. Pedrycz, Data fusion in heterogeneous networks, *Inf. Fusion* 53 (2020) 1–3.
- [80] R. Hamza, Z. Yan, K. Muhammad, P. Bellavista, F.J.I.S. Titouna, A privacy-preserving cryptosystem for IoT E-healthcare, 527, 2020, pp. 493–510.
- [81] M. Sajjad, M. Nasir, F.U.M. Ullah, K. Muhammad, A.K. Sangaiah, S.W.J.I.S. Baik, Raspberry Pi assisted facial expression recognition framework for smart security in law-enforcement services, 479, 2019, pp. 416–431.