Check for updates

IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Multi-stream 3D CNN structure for human action recognition trained by limited data

ISSN 1751-9632
Received on 24th March 2018
Revised 10th November 2018
Accepted on 23rd November 2018
E-First on 28th February 2019
doi: 10.1049/iet-cvi.2018.5088
www.ietdl.org

Vahid Ashkani Chenarlogh[1], *Farbod Razzazi*[1] ✉

[1]*Department of Electrical and Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran*
✉ *E-mail: razzazi@srbiau.ac.ir*

Abstract: Here, the authors proposed a solution to improve the training performance in limited training data case for human action recognition. The authors proposed three different convolutional neural network (CNN) architectures for this purpose. At first, the authors generated four different channels of information by optical flows and gradients in the horizontal and vertical directions from each frame to apply to three-dimensional (3D) CNNs. Then, the authors proposed three architectures, which are single-stream, two-stream, and four-stream 3D CNNs. In the single-stream model, the authors applied four channels of information from each frame to a single stream. In the two-stream architecture, the authors applied optical flow-*x* and optical flow-*y* into one stream and gradient-*x* and gradient-*y* to another stream. In the four-stream architecture, the authors applied each one of the information channels to four separate streams. Evaluating the architectures in an action recognition system, the system was assessed on IXMAS, a data set which has been recorded simultaneously by five cameras. The authors showed that the results of four-stream architecture were better than other architectures, achieving 87.5, 91.66, 91.11, 88.05, and 81.94% recognition rates for cameras 0–4, respectively, using four-stream structure (88.05% recognition rate in average).

1 Introduction

Action recognition aims to recognise the action from one or more visual observations. Human action recognition has been employed in several applications such as behaviour biometrics, video analysis, security, and surveillance, as well as interactive applications such as human–computer interactions or games [1].

Although there is a great tendency to have an end-to-end action recognition system, there are action recognition systems that use handcrafted features proceeded by a classifier [2, 3]. In a study, a number of key frames have been selected in the video stream using the shape flow descriptor [2]. Then, self-similarity matrices have been extracted by computing the similarities between all key frames pairs. In these matrices, diagonal features have been extracted as the high-level features of the action sequence and a support vector machine (SVM) has been used for classification. They evaluated the method using IXMAS data set that has been recorded by five cameras, in which camera 4 is placed on the top of the head and other four cameras are placed horizontally in four orthogonal directions. At the end, they achieved 86.2, 86, 90.03, 84.4, and 81% recognition rates for cameras 0–4 which average accuracy is equal to 85.58% for five views.

Minhas *et al*. [3] have proposed a human action recognition framework using a set of hybrid features which consist of spatio-temporal and local static features that have been extracted using 3D dual-tree complex wavelet transform and affine SIFT local image detector, respectively. They employed extreme learning machine (ELM) as the classifier. They got 99.44% recognition rate on Weizmann data set and 94.83% recognition rate on KTH data set.

In another study, a hierarchical model has been used in a patch-aware representation using discriminative supporting regions [4]. In this approach, a video has been considered as a set of spatio-temporal patches. Each patch indicates the activities of a person. It has been shown that the method is better for close human interaction recognition. They tested their method on BIT, UTI#1, and UTI#2 data sets, which they got 85.38, 93.33, and 91.67% recognition rates, respectively, for each data sets.

Human action recognition in unconstrained videos is a challenging problem that offers outstanding results by using adopting global and local reference points to qualify the motion information [5]. Two kinds of motion reference points have been considered by Jiang *et al*. to decrease the effect of camera movement. Therefore, there is no need to separate foreground from background in the frames. In experiments, they used several databases as test data, including Hollywood2, Olympic sports, HMDB51, and UCF101 data sets. They achieved 65.4, 91.0, 57.3, and 87.2% recognition rates, respectively, in the aforementioned databases. Human action recognition in unconstrained video sequences using a fusion of features has been proposed by Patel *et al*. [6], which consists of different fusion models that use the histogram of oriented gradient features, Haar wavelet transform features, and local binary pattern features in the early fusion scheme, intermediate fusion scheme, and late fusion scheme, respectively. Experiments on UCF11 and ASLAN data sets result in 89.43 and 84.27% recognition rates, respectively.

Although traditional machine learning methods such as SVM or ELM classifiers have been successful for action recognition to a certain extent, these solutions do not perform well in an uncontrolled environment to recognise actions. Traditional algorithms are limited by their representations to model a complex learning task. Recently, due to the increase in computational power of hardware utilities, deep learning algorithms have been proposed to solve traditional artificial intelligence problems. This enables us to extract hidden features of input data and to overcome the traditional algorithms limitations. Convolutional neural networks (CNNs), which are one of the most notable deep learning models, have been widely used for human action recognition. Wang *et al*. have proposed a human action recognition method by applying spatio-temporal features to deep neural networks as inputs. They have constructed a deep neural network using CNN and long short-term memory units, and a temporal attention model [7].

CNNs consist of three types of neural layers, named as convolution layers, pooling layers, and fully connected layers. These multiple layers are trained in a robust manner [8]. Owing to fewer shared connections in the first two types of layers, CNNs are faster from traditional neural networks in the training stage [9]. In video processing, due to the existence of three variables (*x*, *y*, and time), the convolution layer uses a three-dimensional (3D) convolution. Three-dimensional convolutions in the convolutional layer of CNNs compute the features from both spatial and temporal aspects of the input video. The 3D convolution layer connects

© The Institution of Engineering and Technology 2018

multiple sequential temporal frames of the previous layer to the output feature maps of the convolutional layer [10].

CNNs have been used in a variety of applications due to their successful results. Guo *et al*. [11] have proposed a learning approach for a set of effective multiple deep features using a pre-trained CNN model on ImageNet data set. Then, they have fine-tuned the net for object retrieval in surveillance videos. To improve the retrieval efficiency, they have encoded the deep features into 256-bit codes using locality-sensitive hash. Finally, they adopted a late fusion strategy. A recognition framework with an improved spatio-temporal feature extraction technique has been proposed in [12]. The proposed framework has extracted key frames. Then, the multi-features fusion is constructed by available features and max pooling, respectively. Finally, SVM has been used for classification of the human activities using multiple features.

Simonyan and Zisserman [13] tried to learn the spatio-temporal features of videos by two parallel CNN streams, employing spatial and temporal features in separate streams. The streams have been fused finally. The first stream has focused on recognising the objects from video frames, and the other one has been used to recognise the action from the motion in the form of a dense optical flow stream.

In another study, a multi-resolution CNN architecture has been employed to speed up the training of the network with a large-scale data set [14]. A reconfigurable CNN with EM optimisation method has been proposed in [15] for action recognition from RGB-D videos. It has been reported that the performance of 3D convolutional networks is better for spatio-temporal feature learning compared to 2D convolutional networks [16]. The performance of $3 \times 3 \times 3$ convolution kernels has been better than competing architectures for 3D convolutional networks in all layers.

The computational cost of optical flow calculation is very high. Zhang *et al*. [17] have tried to replace optical flow with the motion vector obtained from compressed videos without additional calculations. Owing to inaccurate and noisy motion patterns, they have tried to transfer the optical flow learned data to motion vector in a CNN.

There are previous multi-stream structures to fuse multi-modal information in video and multi-view learning [18, 19]. Zhigang *et al*. have proposed a multi-stream CNN architecture to recognise human actions by encoding appearance, motion, and the temporal video tubes of the human-related regions. They considered human-related regions as the most informative features. Next, they have introduced a foreground detection algorithm to detect the motion of an actor. Then, based on the detected human body, they have constructed an appearance and a motion stream to propose a human-related multi-stream CNN by combining the traditional CNN stream with the novel human-related streams [20].

A multi-stream CNN has been proposed by Tu *et al*. to use semantic-derived multi-modalities in spatial and temporal domains [21]. Since the performance of CNN for action recognition depends on both semantic visual cues and the network architecture, they have proposed a novel spatio-temporal saliency-based video object segmentation model to extract the useful human-related semantics. After estimating the saliency maps, an energy function is constructed to segment two semantic cues: the actor and one distinctive acting part of the actor. Then, they have modified the architecture of the two-stream network to design a multi-stream network, consisting three two-stream networks to extract deeper visual features of multi-modalities in multi-scale spatio-temporal sequence.

Recently, advanced techniques have improved the accuracy by using different streams of data and relying on CNN. Khong *et al*. have tried to propose a method that exploits both RGB and optical flow for human action recognition. They have used a two-stream CNN that takes RGB and optical flow computed from RGB stream as inputs. Each stream works independently. Then the streams are combined by either early fusion or late fusion to recognise the actions. They showed that two-stream CNN outperforms one-stream CNN [22].

Although this study does not use raw video frames, there are a variety of applications that use raw video as the input of CNN to extract the efficient features for a certain application automatically. However, the drawback of this idea is the requirement of large set of samples for the learning algorithm. Although there are some studies that employ features as the CNN input and there are some CNN multi-stream approaches, there is no research on the impact of structure selection for CNN in multi-stream architecture, using different features in different streams in the case of limited data conditions.

In this paper, our contributions can be summarised as follows:

- We provide optflow-*x*, optflow-*y*, gradient-*x*, and gradient-*y* on each of the input frames. We provide different architectures that processes the input features at different streams of single-stream, two-stream, and four-stream CNN architectures.
- Experimental results show that the proposed method has superior results with respect to other approaches on IXMAS data set.

The rest of this paper is organised as follows: we propose a variety of CNN structures in Section 2. Experimental results are introduced in Section 3, and we conclude the paper in Section 4.

2 Proposed multi-stream action recognition architecture

2.1 Motivation

CNNs require a large number of samples to abstract the efficient features automatically. Although CNN can adapt itself with a large number of training samples; however, the large number of sample as a requirement is a limitation, especially when there is limited available training data. Indeed, the size of training set is very limited in video action recognition applications in practice because the procedure of data set annotation and labelling the samples should be carried out by an expert. In addition, the procedure is very time-consuming, expensive, and laborious. Therefore, it is very common to train the action recognition systems with limited data (e.g. IXMAS data set).

Although some studies have proposed to extend the training set by augmentation methods, the method has been just applied for images not videos. Our method is not based on augmentation approach. To overcome the limitation of training data, we propose to help CNN by applying pre-extracted features to CNN. The usage of several streams makes the number of parameters in each CNN small and this is the main philosophy, which supports the learning capability in our multi-stream CNN. Our observations show that single-stream and two-stream features, injected to CNN, degrade the performance in comparison with single-stream raw data and the features should be applied to different CNN architectures. In this section, the idea of multi-stream 3D CNN architecture is proposed and in the next section, the idea is evaluated and compared with traditional architectures.

2.2 Three-dimensional CNN architecture

In the following, we describe a few 3D CNNs architectures that we proposed for human action recognition. In the proposed model, as is shown in Fig. 1, we first applied four hardwired kernels to obtain optflow-*x*, optflow-*y*, gradient-*x*, and gradient-*y* on each of the input frames. The optflow-*x* and optflow-*y* were achieved by calculating optical flows along the horizontal and vertical directions, respectively. In addition, by calculating gradients along the horizontal and vertical directions, we obtained the feature maps of the gradient-*x* and gradient-*y*, respectively. Ji *et al.* [10] introduced optical flow and gradients features to encode raw frames on features and expressed that this scheme usually leads to better performance.

In this paper, this scheme is used to overcome the limitations on CNN training data by providing handcrafted features along with multi-stream structure. Therefore, both extracted features and multi-stream strategy play role in this success.

The Lucas–Kanade algorithm is used to extract optical flow features [23]. Fig. 2 shows these features on one frame. We resampled the input video to 43 frames from each sample and

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

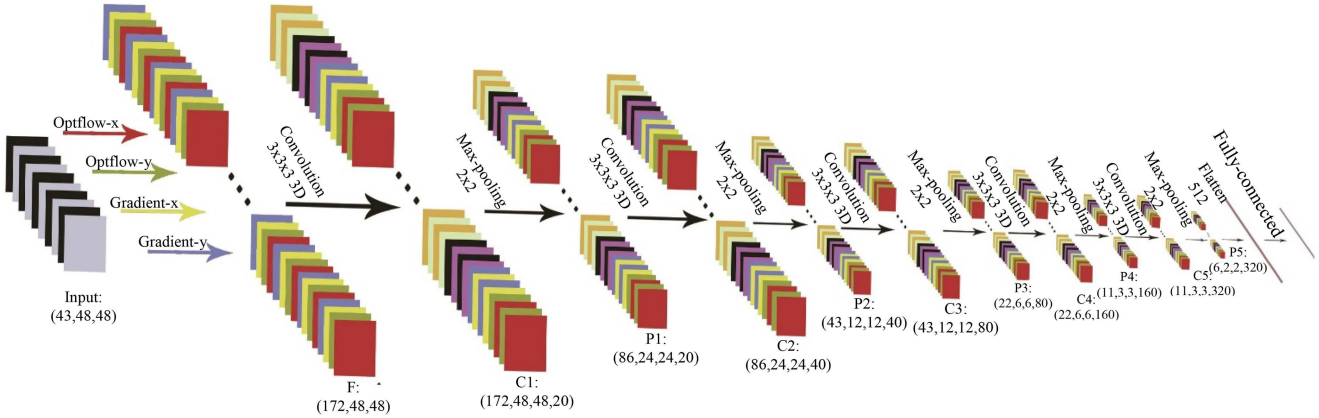

Fig. 1 *Basis 3D CNN model for human action recognition used in the proposed architecture*

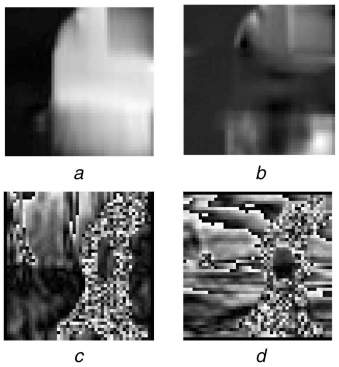

Fig. 2 *Set of features as 3D CNNs inputs by applying four hardwired kernels on one frame of walk action*
(a) Optflow-*x* feature map, ***(b)*** Optflow-*y* feature map, ***(c)*** Gradient-*x* feature map, ***(d)*** Gradient-*y* feature maps

extracted 4 information channels. Therefore, we have 172 channels of information for each sample, to apply to the 3D CNNs. We fixed inputs frame size to 48 × 48 pixels and normalised data by standard deviation as a preprocessing of the input of the network.

As it can be observed in Fig. 1, we have five 3D convolution layers and five pooling layers, which come one after another. After each convolution layer, we used batch normalisation layer [24], which leads to accelerate network training and enables us to use higher learning rates. In addition, rectified linear units function (ReLU) [25] was used as the activation function after this layer. In all layers, we applied 3D convolutions with kernel size of 3 × 3 × 3 on each of the inputs separately. Next, we applied max-pooling layer with 2 × 2 kernel size in all layers on each of the feature maps from previous layers, which leads to the same number of feature maps with reduced spatial resolution.

After multiple convolution and subsampling layers, input frames were converted into a feature vector. We used a fully connected neural network with 512 hidden neurons for final classification. The output layer includes the same number of action classes that are fully connected to each of the 512 units in the feature vector layer. Dropout technique was employed in fully connected layers [26], to reduce the chance of over-fitting. It is worthy to note that designing a robust CNN architecture depends on input data specifications. In the video, to segment the data into distinct informative streams, we tried to separate the video data into static features in X and Y directions and dynamic features in X and Y directions. It seems that different CNN streams can automatically extract informative features from these distinct streams with limited data. Then, these features were fused in the fully connected MLP structure. The number of layers and other parameters of each CNN stream were empirically optimised. Additionally, the number of used layers and the characteristics of convolutional layers or other layers can be changed depending on input data. As a result, the proposed architectures are tried with a variety of layers and the other specifications and the final architecture are achieved empirically.

In all architectures, we used the mentioned CNN model, which is shown in Fig. 1, as the basis of the stream. The final tuned hyper-parameters include 3D convolution parameters as the 3 × 3 × 3 size with padding. We set 20, 40, 80, 160, and 320 filters in five convolution layers, respectively. The filter window was swept on the frame by 1 × 1 × 1 steps in each layer. After batch normalisation and ReLU layers, which is followed by each convolution layer, we used max pooling with the kernel size 2 × 2 with padding. The max-pooling operator was swept on the frame by 2 × 2 steps in five layers. At the end, stochastic gradient descent learning method was used to train the model.

2.3 Architecture variations

We investigated several options for the CNN architecture, which are depicted in Fig. 3. As mentioned above, we used four features maps from each frame to apply to the 3D CNNs. Three architectures have been used that are single-stream, two-stream, and four-stream architectures. In all architectures, the number of convolution and subsampling layers are the same. In this figure, red boxes represent input data, blue boxes represent convolutional layers, yellow boxes indicate max-pooling layers, violet boxes indicate fusion layer, and green boxes show flattening and then passing through a fully connected layer operations, respectively. We first describe a baseline single-stream architecture and then discuss about two and four streams, respectively.

Single-stream architecture: This architecture is similar to the model used in [10]. Instead, we applied four features maps of each frame as 3D CNNs inputs. In this structure, we have one stream that contains five convolutional layers with five max-pooling layers and a fully connected layer. Therefore, we applied all four features on each frame to this stream.

Two-stream architecture: In this structure, we have two streams to train 3D CNNs model that contains five convolutional layers and five max-pooling layers. To this end, we applied optflow-*x* and optflow-*y* to one stream and gradient-*x* and gradient-*y* to another stream. Then, we merged two streams before fully connected layer by concatenation.

Four-stream architecture: The four-stream architecture consists of four streams which we applied each one of four features maps to separate streams separately. This structure, like single and two-stream architectures, contains five convolution and max-pooling layers along with a fully connected layer, respectively. Then, we concatenated the stream outputs before the fully connected layer.

3 Experiments

We conducted the tests on IXMAS data set. IXMAS data set has been prepared for view-invariant human action recognition [27], consisting 12 actors, performing 13 actions in three times. The actions are check watch, cross-arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up, and throw. In order to homogenise the number of classes with other works in the

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

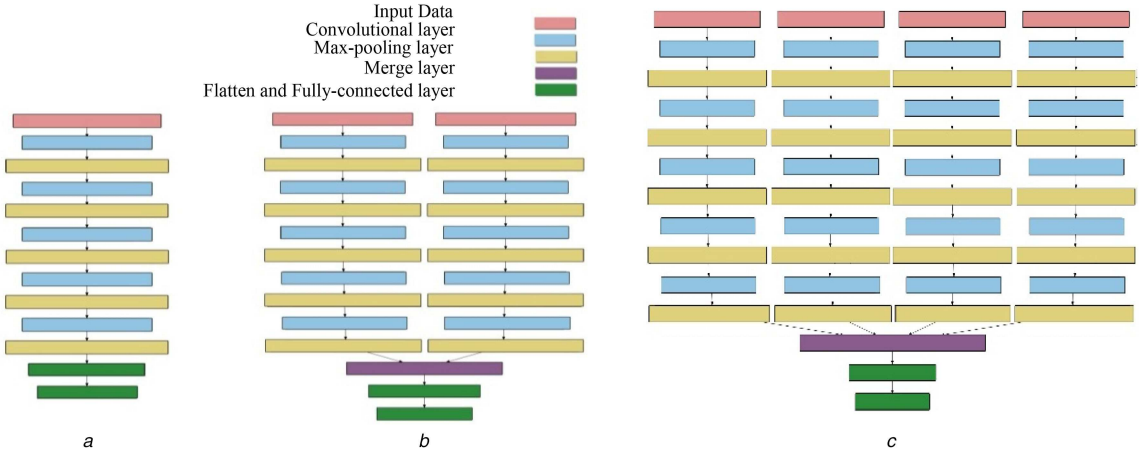

Fig. 3 *Proposed architectures for human action recognition over 3D CNNs*
(a) Single-stream architecture, ***(b)*** Two-stream architecture, ***(c)*** Four-stream architecture

Fig. 4 *Frame samples for sample actions from IXMAS multi-view human action recognition data set*

Table 1 Average recognition rates for each action class after five-fold cross-validation on cameras 1 and 2 on IXMAS data set. 'S', 'T', and 'F' stand for single-stream, two-stream, and four-stream architectures, respectively

Classes	Cam1			Cam2		
Structures	S	T	F	S	T	F
check watch	95	72	93	91	89	88
cross-arms	92	95	92	77	90	90
get up	*100*	*100*	*100*	*100*	*100*	*100*
kick	76	100	95	72	94	88
pick up	*97*	*95*	*96*	*91*	*92*	*91*
point	13	49	42	43	44	49
punch	57	81	75	45	68	73
scratch head	62	82	81	82	79	86
sit down	*100*	*100*	*100*	*100*	*100*	*100*
turn around	*100*	*98*	*100*	*100*	*100*	*100*
walk	*98*	*100*	*100*	*100*	*98*	*98*
wave	84	80	80	85	86	86
average	81.16	87.66	87.83	82.16	86.66	87.41

references, we conducted the tests on 12 action classes without throw action. The data set has been recorded by five cameras in a controlled environment. There are 30 videos per class in the data set. Each video consists of an images sequence with the size of 64 × 48, resulting in totally 1800 videos for all five cameras. Fig. 4 shows some example images from IXMAS data sets. We did not use the multi-view cross-information in a multi-view platform.

Given that the samples in IXMAS data set have different number of frames, we performed a preprocessing to eliminate non-useful frames and carried out down-sample/up-sample on each video to have similar temporal length samples. To up-sample and down-sample the video stream temporally, we used standard uniform interpolation/decimation sampling procedures by Corel video studio software. Then, we chose 43 frames from each sample to apply to the 3D CNNs.

3.1 Experiments on IXMAS data set

To evaluate the effectiveness of our 3D CNN proposed architectures, we reported the result using five-fold cross-validation strategy. Generally, 12 classes from IXMAS data set have 360 samples. Twenty per cent of these samples (72 samples) were used for test set in each fold. The results of action classes after 60 epochs in each fold, for camera 1 and camera 2 views, with all three proposed architectures are tabulated in Table 1. The results of the actions 'get up', 'pick up', 'sit down', 'turn around', and 'walk' were the best results, which are italicised, while the 'point' action has the worst performance. This demonstrates that optical-flow and gradients features in recognition of some actions like 'getting up', 'sitting down', 'turning around', and 'walking' actions perform better than other actions. These actions are very dynamic and the change rate of subsequent frames is significant.

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

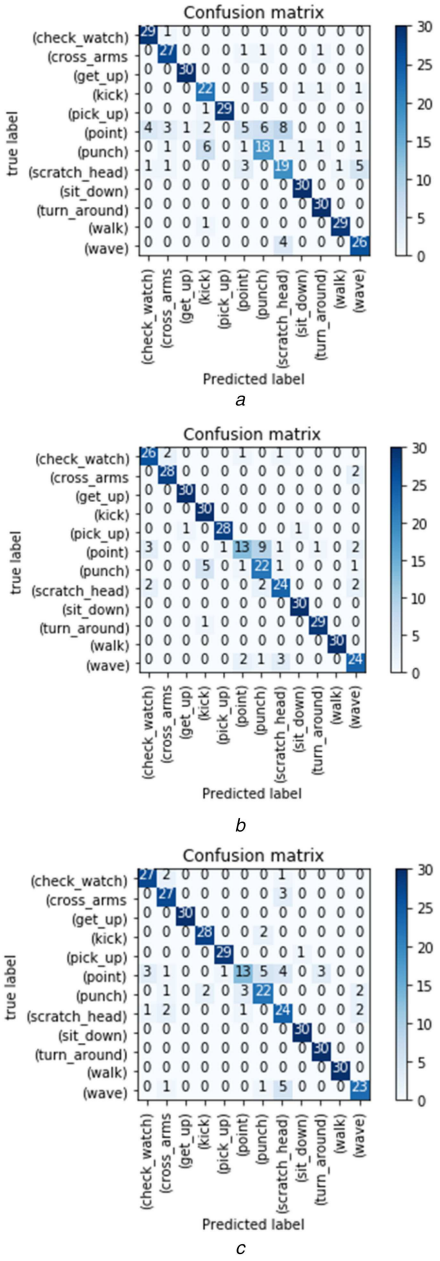

Fig. 5 *Confusion matrices of the averaged five-fold cross-validation on camera 1 on IXMAS data set*

(a) Single-stream, ***(b)*** Two-stream, ***(c)*** Four-stream architectures

The results reveal that the main contribution in the improvement belongs to 'kick' and 'cross-arms' actions in camera 1, and 'check-watch' and 'kick' actions in camera 2. It seems that if the action is simply represented in a camera, there is no need to use a multi-stream architecture. However, in the actions where the representation is complicated in a special view, the multi-stream idea improves the results significantly.

Fig. 5 shows the confusion matrix of the first camera for our three structures after 60 epochs. In this figure, the horizontal axis is the predicted label and the vertical axis is the true label. The obtained results from combined confusion matrixes of five folds show that four streams structure improved the recognition accuracy. In this figure, the worst case is in the 'point' action, where just 13 samples were truly predicted, but 5 samples are wrongly predicted as 'punch' action, 3 samples as 'check-watch' action, 1 sample as 'cross-arms' action, 4 samples as 'scratch-head' action, 3 samples as 'turn-around' action, and 1 sample was wrongly predicted as 'pick-up' action. Generally, as mentioned above, the results show that the worst recognition case is in the point action.

It is interesting to note that the hand-related actions (point, punch, scratch-head, check-watch, cross-arms) were more confused with each other. Especially 'point' was easily misclassified as 'punch'. While the best predictions were on 'get up', 'sit down', 'turn around', and 'walk' that were predicted completely true.

To provide a fair comparison with different state of the art studies, some recent studies were chosen. In [28], human action recognition has been addressed by capturing the structure of temporal similarities and dissimilarities within an action sequence. Self-similarity measure has been proposed for view-independent video analysis. Farhadi *et al.* [29] have introduced a set of latent variables that the appearance feature has been confounded in different viewpoints. They have illustrated that discriminative information of different views might improve the performance. Additionally, they have indicated that the discriminative aspects of views are useful for action classification. Images from different views tend to share discriminative aspects of different objects that have strong appearance similarities. Liu *et al.* [30] have presented an approach to recognise human actions from different views by view knowledge transfer. By modelling an action as a bag of visual words, some high-level features can be shared across view. To discover these features, they have used a bipartite graph to model two view-dependent vocabularies. Then, to co-cluster two vocabularies into visual words, they have applied a bipartite graph partitioning. In [31], Zhen *et al.* have tried to model human actions by explicit coding motion and structure features, which have been separately extracted from the video. To this end, they applied motion template to encode the motion information and then extract the image planes from different frames to capture the structure information. New spatio-temporal context distribution features of interest points for action recognition have been proposed in [32]. In that paper, a global GMM has been learnt using a relative coordinate feature from all training data. To capture the spatio-temporal relationships at different levels, multiple GMM has been utilised. Zhen *et al.* have introduced spatio-temporal Laplacian pyramid coding (STLPC), for holistic representation of human actions. STLPC represents a video sequence as a whole with spatio-temporal feature that has been directly extracted from it. This presents the loss of information in sparse representation [33].

The results of the comparison of the proposed method to these methods on IXMAS data set are shown in Table 2. As it can be observed, the experiment results in four-stream architecture are better than single and two-stream results; because each CNN network just learn one kind of inputs. The highest accuracies belong to cameras 1 and 2, while the least accuracy belongs to camera 4, in which the camera is placed on the top of the head. This is because people have got out of the capture box when performing certain actions like 'walk' and 'turn around' in this view. Thus, some information was lost. However, the action is not guaranteed to be performed in the direction of a specific camera. In addition, due to bad camera stand view, some of actions may be occluded. This causes the accuracy decrease in that view. Two samples from the camera 4 views for 'kick' and 'point' actions are presented in Fig. 6. In addition, we compared our results with respect to other research, which shows that our results outperform most of the previous works.

To show the effect of applying optical-flow and gradients features in horizontal and vertical directions and also to show the effect of our three architectures on human action recognition, we applied raw video frames to our main model in a single-stream CNN architecture.

Table 3 shows the results of this experiment. Human action recognition accuracy in using raw video frames is even more than single-stream and two-stream architectures, where four features have been extracted and employed. This can be concluded that although it can be argued that the use of handcrafted features in limited data conditions; however, if these features are not given to separate streams, the network will not be able to learn well. Consequently, the results were poorer than using the raw frames as the input of the 3D CNNs architecture. In contrast, applying the four features set in a four-stream architecture outperform the raw data input results. This can be interpreted that when the training data is limited, improving the performance requires both

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

modifications; the features as the input data and changing the architecture to a four-stream architecture. This may be due to the need of the learning machine to limit the parameters as the system can be trained.

Furthermore, we performed additional experiments on samples in camera 0 from IXMAS data set on three structures with different number of layers and parameters to show that our main model performs well in this case. We conducted several experiments on this subject, which are presented in Table 4.

Table 2 Accuracy rates in different views on IXMAS data set for single-stream, two-stream, and four-stream architectures with cross-validation method

Model	Cam0	Cam1	Cam2	Cam3	Cam4
Junejo *et al.* [28]	66.5	66.5	66.7	64.8	44.8
Farhadi *et al.* [29]	74	77	76	73	72
Liu *et al.* [30]	79	74.7	75.2	76.4	71.2
Changhong and Zongliang [2]	86.2	86	90.3	84.4	81
Zhen *et al.* [31]	84.9	87.9	90.1	86.9	78.9
Wu *et al.* [32]	81. 9	80.1	77.1	77.6	73.4
Zhen *et al.* [33]	**91.8**	88.8	**91.7**	87.4	81.4
Liu *et al.* [34]	84.7	89.0	85.6	84.5	80.1
single-stream	78.33	83.61	84.16	82.77	75.55
two-stream	85.55	90.28	87.5	85.27	80.55
four-stream	**87.5**	**91.66**	**91.11**	**88.05**	**81.94**

Bold values indicate the best results.

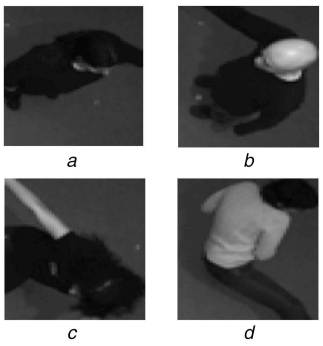

Fig. 6 *Sample frames of camera 4 from IXMAS data set*
(a**, **b) 'Point' action, ***(c**, **d)*** 'Kick' action

In this table, dense indicates the number of hidden neurons in the fully connected layer; convolution kernel represents the kernel size, which is used in the convolutional layer. In this case, $7 \times 5 \times 3$ means that we used $7 \times 7 \times 3$ as the kernel size in the first convolutional layer and $5 \times 5 \times 3$ in the second layer and $3 \times 3 \times 3$ in the third convolutional layer. In addition, 'D' means that we used the dropout layer after each convolutional layer to reduce over-fitting. The number of filters indicates the number of convolution filters used in each layer. In addition, it indicates the number of convolutional layers, which was used. We used max-pooling layer after each convolutional layer and others parameters are similar to our main model, which have been described in Fig. 1.

In all three structures, by decreasing the number of convolutional layers as well as the number of hidden neurons, the network needs more epochs to learn efficiently. In addition, after these experiments, we concluded that the performance of $3 \times 3 \times 3$ convolution kernel size is better than other kernel sizes. In addition, the first and the second rows of Table 4 show that the dropout strategy reduces the accuracy in limited data. This may be due to missing informative data in dropout. Generally, the results show that our main model with the proposed architecture yields the best performance.

4 Conclusion

According to the challenge of training 3D CNNs models using limited data, we proposed an architecture to improve the action recognition performance in such situations. We proposed a 3D CNNs model in three different structures. We used a hardwired layer to extract four features maps from each input frame as the inputs. They were optical flow and gradients in horizontal and vertical directions, which were applied to the 3D CNNs. It seems that in the case of using raw data as the input, the network cannot abstract the important features automatically. Consequently, we applied handcrafted features in separated streams as a solution to help 3D CNN to learn discriminative features.

We constructed our 3D CNNs model in single-stream, two-stream, and four-stream architectures for human action recognition on IXMAS data set. Comparing among these architectures, we showed that in the case of limited data 3D CNN training, the use of handcrafted features in a four-stream architecture improves the recognition performance. Each stream can learn each feature map separately well; while by applying handcrafted features to single-stream and two-stream structures, we achieved poor performance compared to applying raw frames as the input.

Table 3 Comparison of raw frames results versus optflow-*x*, optflow-*y*, gradient-*x*, and gradient-*y* features applied to 3D CNNs in three structures. 'S' (single-stream), 'T' (two-stream), 'F' (four-stream)

	Camera 0			Camera 1			Camera 2			Camera 3			Camera 4		
Structures	S	T	F	S	T	F	S	T	F	S	T	F	S	T	F
raw frames	85.55	—	—	89.16	—	—	89.44	—	—	86.11	—	—	77.5	—	—
features	78.33	85.55	**87.5**	83.61	90.28	**91.66**	84.16	87.5	**91.11**	82.77	85.27	**88.05**	75.55	80.55	**81.94**

Bold values indicate the best results.

Table 4 Comparison of the proposed model with different models in all three proposed structures experimented on camera 0

Model	Number of filters	Convolution kernel	Epochs	Dense	Accuracy
single-stream	[20, 40, 80, 120]	3 × 3 × 3, D	360	256	66.67
single-stream	[10, 20, 40]	3 × 3 × 3	720	160	68.06
single-stream	[10, 20, 40]	3 × 3 × 3	360	256	68.06
two-stream	[10, 20, 40]	3 × 3 × 3	600	256	79.17
two-stream	[10, 20, 40]	7 × 5 × 3	650	180	75
two-stream	[10, 20, 40, 80]	3 × 3 × 3	60	512	80.56
four-stream	[10, 20, 40]	3 × 3 × 3, D	250	300	75
four-stream	[20, 40, 80, 160]	3 × 3 × 3	60	512	83.33
main (single-stream)	[20, 40, 80, 160, 320]	3 × 3 × 3	60	512	**78.33**
main (two-stream)	[20, 40, 80, 160, 320]	3 × 3 × 3	60	512	**85.55**
main (four-stream)	[20, 40, 80, 160, 320]	3 × 3 × 3	60	512	**87.5**

Bold values indicate the best results.

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The results show that our structures outperform the state-of-the-art action recognition systems, tested on IXMAS data set. In addition, we explored the number of layers and other parameters in our model. We are going to test the idea on multi-view action recognition problems in the limited training data case.

5 References

[1] Poppe, R.: 'A survey on vision-based human action recognition', *Image Vis. Comput.*, 2010, **28**, (6), pp. 976–990

[2] Changhong, C., Zongliang, G.: 'Action recognition from a different view', *China Commun.*, 2013, **10**, (12), pp. 139–148

[3] Minhas, R., Baradarani, A., Seifzadeh, S., *et al.*: 'Human action recognition using extreme learning machine based on visual vocabularies', *Neurocomputing*, 2010, **73**, (10–12), pp. 1906–1917

[4] Kong, Y., Fu, Y.: 'Close human interaction recognition using patch-aware models', *IEEE Trans. Image Process.*, 2016, **25**, (1), pp. 167–178

[5] Jiang, Y., Dai, Q., Liu, W., *et al.*: 'Human action recognition in unconstrained videos by explicit motion modeling', *IEEE Trans. Image Process.*, 2015, **24**, (11), pp. 3781–3795

[6] Patel, C., Garg, S., Zaveri, T., *et al.*: 'Human action recognition using fusion of features for unconstrained video sequences', *Comput. Electr. Eng.*, 2018, **70**, pp. 284–301

[7] Wang, L., Xu, Y., Cheng, J., *et al.*: 'Human action recognition by learning spatio-temporal features with deep neural networks', *IEEE Access.*, 2018, **6**, pp. 17913–17922

[8] Guo, Y., Liu, Y., Oerlemans, A., *et al.*: 'Deep learning for visual understanding: a review', *Neurocomputing*, 2016, **187**, pp. 27–48

[9] Lecun, Y., Kavukcuoglu, K., Farabet, C.: 'Convolutional networks and applications in vision'. Proc. Int. Conf. 2010 IEEE Int. Symp. on Circuits and Systems (ISCAS), Paris, France, June 2010, pp. 253–256

[10] Ji, Sh., Xu, W., Yang, M., *et al.*: '3D convolutional neural networks for human action recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (1), pp. 221–231

[11] Guo, H., Wang, J., Lu, H.: 'Multiple deep features learning for object retrieval in surveillance videos', *IET Comput. Vis.*, 2016, **10**, (4), pp. 268–272

[12] Li, J., Mao, X., Chen, L., *et al.*: 'Human interaction recognition fusion multiple features of depth sequences', *IET Comput. Vis.*, 2017, **11**, (7), pp. 560–566

[13] Simonyan, K., Zisserman, A.: 'Two-stream convolutional networks for action recognition in video', *Adv. Neural. Inf. Process. Syst.*, 2014, **24**, pp. 1–9

[14] Karpathy, A., Toderici, G., Shetty, S., *et al.*: 'Large-scale video classification with convolutional neural networks'. Proc. Int. Conf. Vision and Pattern Recognition (CVPR), Columbus, OH, USA, June 2014, pp. 1725–1732

[15] Wang, K., Wang, X., Lin, L., *et al.*: '3D human activity recognition with reconfigurable convolutional neural networks'. Proc. of the 22nd ACM Int. Conf. on Multimedia, Orlando, Florida, USA, November 2014, pp. 97–106

[16] Tran, D., Bourdev, L., Fergus, R., *et al.*: 'Learning spatiotemporal features with 3D convolutional networks'. Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), Santiago, Chile, December 2015, pp. 4489–4497

[17] Zhang, B., Wang, L., Wang, Z., *et al.*: 'Real-time action recognition with enhanced motion vector CNNs'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, June 2016, pp. 2718–2726

[18] Wu, Z., Jiang, Y., Ye, H., *et al.*: 'Fusion multi-stream deep networks for video classification'. Computer Vision and Pattern Recognition (cs.CV); Multimedia, November 2015, pp. 1–9

[19] He, T., Mao, H., Yi, Z.: 'Moving object recognition using multi-view three-dimensional convolutional neural networks', *Neural Comput. Appl.*, 2017, **28**, (12), pp. 3827–3835

[20] Zhigang, T., Wei, X., Qianging, Q., *et al.*: 'Multi-stream CNN: learning representations based on human-related regions for action recognition', *Pattern Recognit.*, 2018, **79**, (7), pp. 32–43

[21] Tu, Z., Xie, W., Dauwels, J., *et al.*: 'Semantic cues enhanced multi-modality multi-stream CNN for action recognition', *IEEE Trans. Circuits Syst. Video Technol.*, 2018, Early access, doi: 10.1109/TCSVT.2018.2830102

[22] Khong, V.-M., Tran, T.-H.: 'Improving human action recognition with two-stream 3D convolutional neural network'. Proc. of the 2018 1st Int. Conf. on Multimedia Analysis and Pattern Recognition (MAPR), Ho chi minh city, Vietnam, April 2018, pp. 1–6

[23] Aires, K.R.T., Santana, A.M., Medeiros, A.A.D.: 'Optical flow using color information: preliminary results'. Proc. of the 2008 ACM Symp. on Applied Computing, Fortaleza, Ceara, Brazil, 2008, pp. 1607–1611

[24] Ioffe, S., Szegedy, C.: 'Batch normalization: accelerating deep network training by reducing internal covariate shift'. Proc. of the 32nd Int. Conf. on Machine Learning, PMLR 37, Lille, France, July 2015, pp. 448–456

[25] Dahl, G.E., Sainath, T.N., Hinton, G.E.: 'Improving deep neural networks for LVCSR using rectified linear units and dropout'. Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, May 2013, pp. 8609–8613

[26] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'Imagenet classification with deep convolutional neural networks', *Adv. Neural. Inf. Process. Syst.*, 2012, **25**, pp. 1–9

[27] Holte, M.B., Tran, C., Trivedi, M.M., *et al.*: 'Human pose estimation and activity recognition from multi-view videos: comparative explorations of recent developments', *IEEE J. Sel. Top. Signal. Process.*, 2012, **6**, (5), pp. 538–552

[28] Junejo, I.N., Dexter, E., Laptev, I., *et al.*: 'View independent action recognition from temporal self-similarities', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (1), pp. 172–185

[29] Farhadi, A., Kamali Tabrizi, M., Endres, I., *et al.*: 'A latent model of discriminative aspect'. Proc. of the IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, October 2009, pp. 948–955

[30] Liu, J., Shah, M., Kuipers, B., *et al.*: 'Cross-view action recognition via view knowledge transfer'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, June 2011, pp. 3209–3216

[31] Zhen, X., Shao, L., Tao, D., *et al.*: 'Embedding motion and structure feature for action recognition', *IEEE Trans. Circuits Syst. Video Technol.*, 2013, **23**, (7), pp. 1182–1190

[32] Wu, X., Xu, D., Duan, L., *et al.*: 'Action recognition using context and appearance distribution features'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, June 2011, pp. 489–496

[33] Zhen, X., Shao, L., Tao, D., *et al.*: 'Spatio-temporal Laplacian pyramid coding for action recognition', *IEEE Trans. Cybernet.*, 2014, **44**, (6), pp. 817–827

[34] Liu, L., Shao, L., Rockett, P.: 'Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition', *Pattern Recognit.*, 2013, **46**, (7), pp. 1810–1818

© The Institution of Engineering and Technology 2018

17519640, 2019, 3, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cvi.2018.5088 by Test, Wiley Online Library on [13/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License