Dividing and Aggregating Network for Multi-view Action Recognition

Dongang Wang[1][0000−0001−5805−0244], Wanli Ouyang[1,2][0000−0002−9163−2761],
Wen Li[3][0000−0002−5559−8594], and Dong Xu[1]

[1] The University of Sydney, School of Electrical and Information Engineering
{dongang.wang,wanli.ouyang,dong.xu}@sydney.edu.au
[2] The University of Sydney, SenseTime Computer Vision Research Group
[3] ETH Zurich, Computer Vision Laboratory
liwen@vision.ee.ethz.ch

Abstract. In this paper, we propose a new Dividing and Aggregating Network (DA-Net) for multi-view action recognition. In our DA-Net, we learn view-independent representations shared by all views at lower layers, while we learn one view-specific representation for each view at higher layers. We then train view-specific action classifiers based on the view-specific representation for each view and a view classifier based on the shared representation at lower layers. The view classifier is used to predict how likely each video belongs to each view. Finally, the predicted view probabilities from multiple views are used as the weights when fusing the prediction scores of view-specific action classifiers. We also propose a new approach based on the conditional random field (CRF) formulation to pass message among view-specific representations from different branches to help each other. Comprehensive experiments on two benchmark datasets clearly demonstrate the effectiveness of our proposed DA-Net for multi-view action recognition.

Keywords: Dividing and Aggregating Network · multi-view action recognition · large-scale action recognition.

1 Introduction

Action recognition is an important problem in computer vision due to its broad applications in video content analysis, security control, human-computer interface, etc. Recently, significant improvements have been achieved, especially with the deep learning approaches [27,24,35,23,40].

Multi-view action recognition is a more challenging task as action videos of the same person are captured by cameras from different viewpoints. It is well-known that failure in handling feature variations caused by viewpoints may yield poor recognition results [42,43,31].

One motivation of this paper is to learn view-specific deep representations. This is different from existing approaches for extracting view-invariant features using global codebooks [28,18,19] or dictionaries [43]. Because of the large divergence in specific settings of viewpoint, the visible regions are different, which

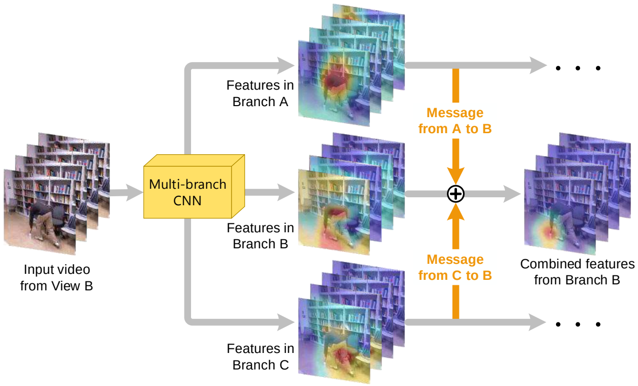


Fig. 1. The motivation of our work for learning view-specific deep representations and passing messages among them. The features extracted in different branches should focus on different regions related to the same action. Message passing from different branches will help each other and thus improve the final classification performance. We only show the message passing from other branches to Branch B for better illustration.

makes it difficult to learn invariant features among different views. Thus, it is more beneficial to learn view-specific feature representation to extract the most discriminative information for each view. For example, at camera view A, the visible region could be the upper part of human body, while the camera views B and C have more visible cues like hands and legs. As a result, we should encourage the features of videos captured from camera view A to focus on the upper body region, while the features of videos from camera view B to focus on other regions like hands and legs. In contrast, the existing approaches tend to discard such view-specific discriminative information.

Another motivation of this paper is that the view-specific features can be used to help each other. Since these features are specific to different views, they are naturally complementary to each other. This provides us with the opportunity to pass message among these features so that they can help each other through interaction. Take Fig. 1 as an example, for the same input image from View B, the features from branches A, B, C focus on different regions. By conducting well-defined message passing, the specific features from View A and View C can be used for refining the features for View B, leading to more accurate representations for action recognition.

Based on the above two motivations, we propose a *Dividing and Aggregating Network (DA-Net)* for multi-view action recognition. In our DA-Net, each branch learns a set of view-specific features. We also propose a new approach based on *conditional random field* (CRF) to learn better view-specific features by passing message to each other. Finally, we introduce a new fusion approach by using the predicted view probabilities as the weights for fusing the classification results from multiple view-specific classifiers to output the final prediction score for action classification.

To summarize, our contributions are three-fold:

1) We propose a multi-branch network for multi-view action recognition. In this network, the lower CNN layers are shared to learn view-independent representations. Taking the shared features as the input, each view has its own CNN branch to learn its view-specific features.

2) Conditional random field (CRF) is introduced to pass message among view-specific features from different branches. A feature in a specific view is considered as a continuous random variable and passes message to the feature in another view. In this way, view-specific features at different branches communicate and help each other.

3) A new view-prediction-guided fusion method for combining action classification scores from multiple branches is proposed. In our approach, we simultaneously learn multiple view-specific classifiers and the view classifier. An action prediction score is obtained for each branch, and multiple action prediction scores are fused by using the view prediction probabilities as the weights.

2 Related works

Action recognition.Researchers have made significant contributions in designing effective features as well as classifiers for action recognition [17,30,36,34,26]. Wang *et al.* [32] proposed the iDT feature to encode the information from edge, flow and trajectory. The iDT feature became dominant in the THUMOS 2014 and 2015 challenges [7]. In the deep learning community, Tran *et al.* proposed C3D [27], which designs a 3D CNN model for video datasets by combining appearance features with motion information. Sun *et al.* [25] applied the factorization methods to decompose 3D convolution kernels and used the spatio-temporal features in different layers of CNNs. The recent trend in action recognition follows two-stream CNNs. Simonyan and Zisserman [24] first proposed the two-stream CNN to extract features from the RGB key frames and the optical flow channel. Wang *et al.* [34] integrated the key factors from iDT and CNN and achieved significant performance improvement. Wang *et al.* also proposed the temporal segment network(TSN) [35] to utilize segments of videos under the two-stream CNN framework. Researchers also transform the two-stream structure to the multi-branch structure. In [6], Feichtenhofer *et al.* proposed a single CNN that fuses the spatial and temporal features before the final layers, which achieves excellent results. Wang *et al.* proposed a multi-branch neural network, where each branch deals with different level of features and then fuse them together [36]. However, these works did not take the multi-view action recognition into consideration. Therefore, they do not learn view-specific features or use view prediction probabilities as the prior when fusing the classification scores from multiple branches as in our work. They do not use message passing to improve their features, either.

Multi-view action recognition. For the multi-view action recognition tasks where the videos are from different viewpoints, the existing action recognition approaches may not achieve satisfactory recognition results [42,31,15,16].

The methods using view-invariant representations are popular for multi-view action recognition. Wu *et al.* [37] and Turaga *et al.* [28] proposed to construct the common space as the multi-view action feature space by using global GMM or Grassmann and Stiefel manifolds and achieved promising results. In recent works, Zheng *et al.* [43], Kong *et al.* [10] and Hossein *et al.* [19] designed different methods to learn the global codebook or dictionary to better extract view-invariant representations from action videos. By treating the problem as a domain adaptation problem, Li *et al.* [12] and Mancini *et al.* [14] proposed new approaches to learn robust classifiers or domain-invariant features. Different from these methods for learning view-invariant features in the common space, we directly learn view-specific features by using multi-branch CNNs. With these view-specific features, we exploit the relationship among them in order to effectively leverage multi-view features.

Conditional Random Field (CRF). CRF has been exploited for action recognition in [29] as it can connect features and outputs, especially for temporal signals like actions. Chen *et al.* proposed L-CORF [3] for locating actions in videos, where CRF was used for modeling spatial-temporal relationship in each single-view video. CRF could also exploit the relationship among spatial features. It has been successfully introduced for image segmentation in the deep learning community by Zheng *et al.* [44], which deals with the relationship among pixels. Xu *et al.* [39,38] modeled the relationship of pixels to learn the edges of objects in images. Recently, Chu *et al.* [4,5] have utilized discrete CRF in CNN for human pose estimation. Our work is the first for action recognition by exploiting the relationship among features from videos captured by cameras from different viewpoints. Our experiments demonstrate the effectiveness of our message passing approach for multi-view action recognition.

3 Multi-View Action Recognition

3.1 Problem Overview

In the multi-view action recognition task, each sample in the training or test set consists of multiple videos captured from different viewpoints. The task is to train a robust model by using those multi-view training videos, and perform action recognition on multi-view test videos.

Let us denote the training data as $\{(\mathbf{x}_{i,1},\ldots,\mathbf{x}_{i,v},\ldots,\mathbf{x}_{i,V})|_{i=1}^{N}\}$, where $\mathbf{x}_{i,v}$ is the i-th training sample/video from the v-th view, V is the total number of views, and N is the number of multi-view training videos. The label of the i-th multi-view training video $(\mathbf{x}_{i,1},\ldots,\mathbf{x}_{i,V})$ is denoted as $y_i \in \{1,\ldots,K\}$ where K is the total number of action categories. For better presentation, we may use $\mathbf{x}_i$ to represent one video when we do not care about which specific view each video comes from, where $i = 1,\ldots,NV$.

To effectively cope with the multi-view training data, we design a new multi-branch neural network. As shown in Fig. 2, this network consists of three modules. (1) **Basic Multi-branch Module**: This network extracts the common

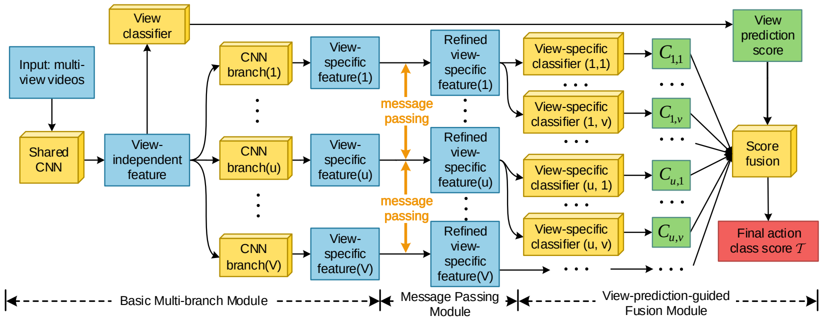

View classifier
View prediction score
Input: multi-view videos
Shared CNN
View-independent feature
CNN branch(1)
CNN branch(u)
CNN branch(V)
View-specific feature(1)
View-specific feature(u)
View-specific feature(V)
message passing
message passing
Refined view-specific feature(1)
Refined view-specific feature(u)
Refined view-specific feature(V)
View-specific classifier (1,1)
View-specific classifier (1, v)
View-specific classifier (u, 1)
View-specific classifier (u, v)
$C_{1,1}$
$C_{1,v}$
$C_{u,1}$
$C_{u,v}$
Score fusion
Final action class score $\mathcal{T}$
Basic Multi-branch Module
Message Passing Module
View-prediction-guided Fusion Module

Fig. 2. Network structure of our newly proposed Dividing and Aggregating Network (DA-Net). (1) **Basic multi-branch module** is composed of one shared CNN and several view-specific CNN branches. (2) **Message passing module** is introduced between every two branches and generate the refined view-specific features. (3) In the **view-prediction-guided fusion module**, we design several view-specific action classifiers for each branch. The final scores are obtained by fusing the results from all action classifiers, in which the view prediction probabilities from the view classifier are used as the weights.

features (*i.e.* view-independent features) for all videos by using one shared CNN, and then extracts view-specific features by using multiple CNN branches, which will be described in Section 3.2. (2) **Message Passing Module**: Based on the basic multi-branch module, we also propose a message passing approach to improve view-specific features from different branches, which will be introduced in Section 3.3. (3) **View-prediction-guided Fusion Module**: The refined view-specific features from different branches are passed through multiple view-specific action classifiers and the final scores are fused with the guidance of probabilities from the view classifier that is trained based on view-independent features.

3.2 Basic Multi-branch Module

As shown in Fig. 2, the basic multi-branch module consists of two parts: 1) *shared CNN*: Most of the convolutional layers are shared to save computation and generate the common features (*i.e.* view-independent features); 2) *CNN branches*: Following the shared CNN, we define V view-specific branches, and view-specific features can be extracted from these branches.

In the initial training phase, each training video $\mathbf{x}_i$ first flows through the shared CNN, and then only goes to the v-th view-specific branch. Then, we build one view-specific classifier to predict the action label for the videos from each view. Since each branch is trained by using training videos from a specific viewpoint, each branch captures the most informative features for its corresponding view. Thus, it can be expected that the features from different views are complementary to each other for predicting the action classes. We refer to this structure as the *Basic Multi-branch Module*.

3.3 Message Passing Module

To effective integrate different view-specific branches for multi-view action recognition, we further exploit the inter-view relationship by using a *conditional random field (CRF)* model to pass message among features extracted from different branches.

Let us denote the multi-branch features for one training video as $\mathbf{F} = \{\mathbf{f}_v\}_{v=1}^{V}$, where each $\mathbf{f}_v$ is the view-specific feature vector extracted from the v-th branch. Our objective is to estimate the refined view-specific feature $\mathbf{H} = \{\mathbf{h}_v\}_{v=1}^{V}$. As shown in Fig. 3(a), we formulate this problem under the CRF framework, in which we learn a new feature representation $\mathbf{h}_v$ for each $\mathbf{f}_v$, and also regularize different $\mathbf{h}_v$'s based on their pairwise relationship. Specifically, the energy function in CRF is defined as,

$$E(\mathbf{H}, \mathbf{F}, \Theta) = \sum_{v} \phi(\mathbf{h}_v, \mathbf{f}_v) + \sum_{u,v} \psi(\mathbf{h}_u, \mathbf{h}_v), \tag{1}$$

in which ϕ is the unary potential and ψ is the pairwise potential. In particular, $\mathbf{h}_v$ should be similar to $\mathbf{f}_v$, namely the refined view-specific feature representation does not change too much from the original representation. Therefore, the unary potential is defined as follows,

$$\phi(\mathbf{h}_v, \mathbf{f}_v) = -\frac{\alpha_v}{2} \|\mathbf{h}_v - \mathbf{f}_v\|^2, \tag{2}$$

where α_v is a weight parameter that will be learnt during the training process. Moreover, we employ a bilinear potential function to model the correlation among features from different branches, which is defined as

$$\psi(\mathbf{h}_u, \mathbf{h}_v) = {\mathbf{h}_v}^{\top} \mathbf{W}_{u,v} \mathbf{h}_u, \tag{3}$$

where $\mathbf{W}_{u,v}$ is the matrix modeling the relationship among different features. $\mathbf{W}_{u,v}$ can be learnt during the training process.

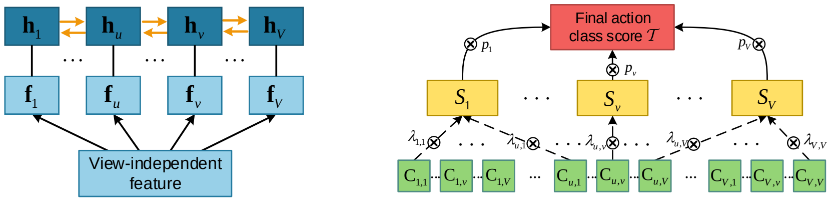
Following [20], we use mean-field update to infer the mean vector of $\mathbf{h}_u$ as:

$$\mathbf{h}_v = \frac{1}{\alpha_v} (\alpha_v \mathbf{f}_v + \sum_{u \neq v} (\mathbf{W}_{u,v} \mathbf{h}_u)). \tag{4}$$

Thus, the refined view-specific feature representation $\{\mathbf{h}_v|_{v=1}^{V}\}$ can be obtained by iteratively applying the above equation.

From the definition of CRF, the first term in Eqn.(4) serves as the unary term for receiving the information from the feature $\mathbf{f}_v$ for its own view v. The second term is the pair-wise term that receives the information from other views u for $u \neq v$. The $\mathbf{W}_{u,v}$ in Eqn.(3) and Eqn.(4) models the relationship between the feature vector $\mathbf{h}_u$ from the u-th view and the feature $\mathbf{h}_v$ from the v-th view.

The above CRF model can be implemented in neural networks as shown in [44,5], thus it can be naturally integrated in the basic multi-branch network, and optimized in the end-to-end training process based on the basic multi-branch



(a) Message passing module

(b) View-prediction-guided fusion module

Fig. 3. The details for (a) inter-view message passing module discussed in Section 3.3, and (b) view-prediction-guided fusion module described in Section 3.4. Please see the corresponding sections for the detailed definitions and descriptions.

module. The basic multi-branch module together with the message passing module is referred to as the *Cross-view Multi-branch Module* in the following sections. The message passing process can be conducted multiple times with the shared $\mathbf{W}_{u,v}$'s in each iteration. In our experiments, we perform only one iteration as it already provides good feature representations.

3.4 View-prediction-guided Fusion

In multi-view action recognition, a body movement might be captured from more than one viewpoint and should be recognized from different aspects, which implies that different views contain certain complementary information for action recognition. To effectively capture such cross-view complementary information, we therefore propose a *View-prediction-guided Fusion Module* to automatically fuse the prediction scores from all view-specific classifiers for action recognition.

Learning view-specific classifiers In the cross-view multi-branch module, instead of passing each training video into only one specific view as in the basic multi-branch module, we feed each video $\mathbf{x}_i$ into all V branches.

Given a training video $\mathbf{x}_i$, we will extract features from each branch individually, which will lead to V different representations. Considering we have training videos from V different views, there would be in total $V \times V$ types of cross-view information, each corresponding to a branch-view pair (u, v) for $u, v = 1, \ldots, V$, where u is the index of the branch and v is the index of the view that the videos belong to.

Then, we build view-specific action classifiers in each branch based on different types of visual information, which leads to $V \times V$ different classifiers. Let us denote $C_{u,v}$ as the score generated by using the v-th view-specific classifier from the u-th branch. Specifically, for the video $\mathbf{x}_i$, the score is denoted as $C^i_{u,v}$. As shown in Fig. 3(b), the fused score of all the results from the v-th view-specific classifiers in all branches is denoted as S_v. Specifically, for the video $\mathbf{x}_i$, the fused score S_v^i can be formulated as follows,

$$S_v^i = \sum_u \lambda_{u,v} C_{u,v}^i, \tag{5}$$

where $\lambda_{u,v}$'s are the weights for fusing $C_{u,v}$'s, which can be jointly learnt during the training procedure and shared by all videos. For the v-th value in the u-th branch, we initialize the value of $\lambda_{u,v}$ when $u = v$ twice as large as the value of $\lambda_{u,v}$ when $u \neq v$, as $C_{v,v}$ is the most related score for the v-th view when compared with other scores $C_{u,v}$'s ($u \neq v$).

Soft ensemble of prediction scores Different CNN branches share common information and have each own refined view-specific information, so the combination of results from all branches should achieve better classification results. Besides, we do not want to use the view labels of input videos during the training or testing process. In that case, we further propose a strategy to fuse all view-specific action prediction scores $\{S_v|_{v=1}^V\}$ based on the view prediction probabilities of each video, instead of using only the one score from the known view as in the basic multi-branch module.

Let us assume each training video $\mathbf{x}_i$ is associated with V view prediction probabilities $\{p_v^i|_{v=1}^V\}$, where each p_v^i denotes the probability of $\mathbf{x}_i$ belonging to the v-th view and $\sum_v p_v^i = 1$. Then, the final prediction score $\mathcal{T}^i$ can be calculated as the weighted mean of all view-specific scores based on the corresponding view prediction probabilities,

$$\mathcal{T}^i = \sum_{v=1}^V p_v^i S_v^i. \tag{6}$$

To obtain the view prediction probabilities, as shown in Fig. 2, we additionally train a *view classifier* by using the common features (*i.e.* view-independent feature) after the *shared CNN*. We use the cross entropy loss for the view classifier and the action classifier, denoted as $\mathcal{L}_{view}$ and $\mathcal{L}_{action}$ respectively.

The final model is learnt by jointly optimizing the above two losses, *i.e.*,

$$\mathcal{L} = \mathcal{L}_{action} + \mathcal{L}_{view}, \tag{7}$$

where we treat the two losses equally and this setting leads to satisfactory results.

The cross-view multi-branch module with view-prediction-guided fusion module forms our *Dividing and Aggregating Network (DA-Net)*. It is worth mentioning that we only use view labels for training the basic multi-branch module and the fine-tuning steps after the basic multi-branch module and the test stages do not require view labels of videos. Even the test video comes from an unseen view, our model can still automatically calculate its view prediction probabilities by using the view classifier, and ensemble the prediction scores from view-specific classifiers for final prediction (see our experiments on *cross-view* action recognition in Section 4.3).

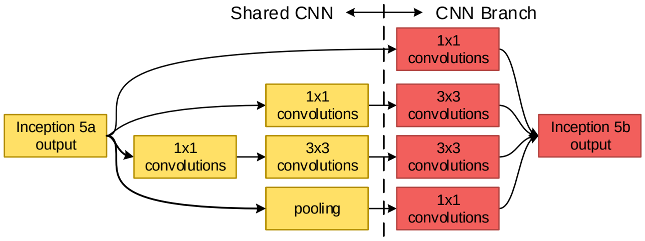


Fig. 4. The layers used in the shared CNN and CNN branches in the `inception_5b` block. The layers in yellow color are included in the shared CNN, while the layers in red color are duplicated for different branches. The layers after `inception_5b` are also duplicated. The `ReLU` and `BatchNormalization` layers after each convolutional layer are treated similarly as the corresponding convolutional layers.

3.5 Network Architecture

We illustrate the architecture of our DA-Net in Fig. 2. The shared CNN can be any of the popular CNN architectures, which is followed with V view-specific branches, each corresponding to one view. Then, we build $V \times V$ view-specific classifiers on top of those view-specific branches, where each branch is connected to V classifiers. Those $V \times V$ view-specific classifiers are further ensembled to produce V branch-level scores using Eqn.(5). Finally, those V branch-level scores are reweighed to obtain the final prediction score, where the weights are the view probabilities generated from the view classifier, which is trained after the shared CNN. Like other deep neural networks, our proposed model can be trained by using popular optimization approaches such as the stochastic gradient descent (SGD) algorithm. We first train the basic multi-branch module to learn view-specific feature in each branch, and then we fine-tune all the modules.

In our implementation, we build our network based on the temporal segment network(TSN) [35] with some modifications. In particular, we use the BN-Inception [9] as the backbone network. The shared CNN layers include the ones from the input to the block `inception_5a`. As shown in Fig. 4, for each path within the `inception_5b` block, we duplicate the last convolutional layer (shown in red in Fig. 4) for multiple times for multiple branches and the previous layers are shared in the shared CNN. The rest average pooling and fully connected layers after the `inception_5b` block are also duplicated for multiple branches. The corresponding parameters are also duplicated at initialization stage and learnt separately. Similarly as in TSN, we also train a two-stream network [24], where two streams are learnt separately using two modalities, RGB and dense optical flow, respectively. In the testing phase, given a test sample with multiple views of videos, $(\mathbf{x}_1, \ldots, \mathbf{x}_V)$, we pass each video $\mathbf{x}_v$ to two streams, and obtain its prediction by fusing the outputs from two streams.

The training of our DA-Net has the same starting point of TSN. We first train the network based on the basic multi-branch module to learn the basic features of each branch and then fine-tune the learnt network by additionally adding the message passing module and view-prediction-guided fusion module.

4 Experiments

In this section, we conduct experiments to evaluate our proposed model by using two benchmark multi-view action datasets. We conduct experiments on two settings: 1) the *cross-subject* setting, which is used to evaluate the effectiveness of our proposed model for learning from multi-view videos, and 2) the *cross-view* setting, which is used to evaluate the generalization ability of our proposed model to unseen views.

4.1 Datasets and Setup

NTU RGB+D (NTU) [21] is a large scale dataset for human action recognition, which contains 60 daily actions performed by 40 different subjects. The actions are captured by Kinect v2 in three viewpoints. The modalities of data including RGB videos, depth maps and 3D joint information, where only the RGB videos are used for our experiments. The total number of RGB videos is $56,880$ containing more than 4 million frames.

Northwestern-UCLA Multiview Action (NUMA)[33] is another popular multi-view action recognition benchmark dataset. In this dataset, 10 daily actions are performed by 10 subjects for several times, which are captured by three static cameras. In total, the dataset consists of $1,475$ RGB videos and the correlated depth frames and skeleton information, where only the RGB videos are used for our experiments.

4.2 Experiments on Multi-view Action Recognition

The *cross-subject* evaluation protocol is used in this experiment. All action videos of a few subjects from all views are selected as the training set, and the action videos of the remaining subjects are used for testing.

For the NTU dataset, we use the same cross-subject protocol as in [21]. We compare our proposed method with a wide range of baselines, among which the work in [21,22,1] include 3D joint information, and the work in [2,13] used RGB videos only. We also include the TSN method [35] as a baseline for comparison, which can be treated as a special case of our DA-Net without explicitly exploiting the multi-view information in training videos. The results are shown in the third column of Table 1. We observe that the TSN method achieves much better results than the previous works using multi-modality data, which could be attributed to the usage of deep neural networks for learning effective video representations. Moreover, the recent works from Baradel *et al.* [2] and Luvizon *et al.* [13] reported the results using only RGB videos, where the work from Luvizon *et al.* [13] achieves similar performance as the TSN method. Our proposed DA-Net outperforms all existing state-of-the-art algorithms and the baseline TSN method.

Table 1. Accuracy comparison between our DA-Net and other state-of-the-art works on the NTU dataset. When using RGB videos, our DA-Net, TSN [35] and the work from Zolfaghari *et al.* [45] use optical flow generated from RGB videos while the rest works do not extract optical flow features. Four methods additionally utilize the pose modality. The best results are shown in bold.

Methods	Modalities	Cross-Subject Accuracy	Cross-View Accuracy
DSSCA-SSLM [22]	Pose+RGB	74.9%	-
STA-Hands [1]	Pose+RGB	82.5%	88.6%
Zolfaghari *et al.* [45]	Pose+RGB	80.8%	-
Baradel *et al.* [2]	Pose+RGB	84.8%	90.6%
Luvizon *et al.* [13]	RGB	84.6%	-
TSN [35]	RGB	84.93%	85.36%
DA-Net(Ours)	RGB	**88.12%**	**91.96%**

Table 2. Average accuracy comparison (the cross-subject setting) between our DA-Net and other works on the NUMA dataset. The results are generated by averaging the accuracy of each subject. The best result is shown in bold.

Methods	Average Accuracy
Li and Zickler [11]	50.7%
MST-AOG [33]	81.6%
Kong *et al.* [10]	81.1%
TSN [35]	90.3%
DA-Net(ours)	**92.1%**

For the NUMA dataset, we use the 10-fold evaluation protocol, where videos of each subject will be used as the test videos each time. To be consistent with other works, we report the video-level accuracy, in which the videos of each view are evaluated separately. The average accuracies are shown in Table 2, where our proposed DA-Net again outperforms all other baseline methods.

The results on both datasets clearly demonstrate the effectiveness of our DA-Net for learning deep models using multi-view RGB videos. By learning view-specific features as well as classifiers and conducting message passing, videos from multiple views are utilized more effectively. As a result, we can learn more discriminative features and our DA-Net can achieve better action classification results when compared with previous methods.

4.3 Generalization to Unseen Views

Our DA-Net can also be readily used for generalization to unseen views, which is also known as the *cross-view* evaluation protocol. We employ the *leave-one-view-out* strategy in this setting, in which we use videos from one view as the test set, and employ videos from the remaining views for training our DA-Net.

Table 3. Average accuracy comparison on the NUMA dataset [33] (the cross-view setting) when the videos from two views are used for training and the videos from the remaining view are used for testing. The best results are shown in bold. For fair comparison, we only report the results from the methods using RGB videos.

{Source}|Target	{1,2}|3	{1,3}|2	{2,3}|1	Average Accuracy
DVV [41]	58.5%	55.2%	39.3%	51.0%
nCTE [8]	68.6%	68.3%	52.1%	63.0%
MST-AOG [33]	-	-	-	73.3%
NKTM [18]	75.8%	73.3%	59.1%	69.4%
R-NKTM [19]	78.1%	-	-	-
Kong *et al.* [10]	-	-	-	77.2%
TSN [35]	84.5%	80.6%	76.8%	80.6%
DA-Net(ours)	**86.5%**	**82.7%**	**83.1%**	**84.2%**

Different from the training process under the cross-subject setting, the total number of branches in the network is set to the total number of views minus 1, since videos from one viewpoint are reserved for testing. During the testing stage, the videos from the target view (*i.e.* unseen view) will go through all the branches and the view classifier can still provide the prediction scores of each testing video belonging to a set of source views (*i.e.* seen views). The scores indicate the similarity between the videos from the target view and those from the source views, based on which we can still obtain the weighted fusion scores that can be used for classifying videos from the target view.

For the NTU dataset, we follow the original cross-view setting in [21], in which videos from view 2 and view 3 are used for training while videos from view 1 are used for testing. The results are shown in the fourth column of Table 1. On this cross-view setting, our DA-Net also outperforms the existing methods by a large margin.

For the NUMA dataset, we conduct three-fold cross validation. The videos from two views together with their action labels are used as the training data to learn the network and the videos from the remaining view are used for testing. The videos from the unseen view are not available during the training stage. We report our results in Table 3, which shows our DA-Net achieves the best performance. Our results are even better than the methods that use the videos from the unseen view as unlabeled data in [10]. The detailed accuracy for each class is shown in Fig. 5. Again we observe that DA-Net is better than nCTE [8] and NKTM [18] in almost all the action classes.

From the results, we observe that our DA-Net is robust even without using videos from the target view during the training process. A possible explanation is as follows. Building upon the TSN architecture, our DA-Net further learns view-specific features, which produces better representations to capture information from each view. Second, the message passing module further improves the feature representation on different views. Finally, the newly proposed soft ensemble

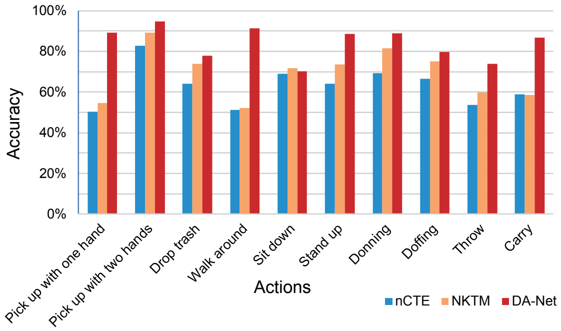


Fig. 5. Average recognition accuracy in each class on the NUMA dataset under the cross-view setting. All the three methods do not utilize the features from the unseen view during the training process.

fusion scheme using view prediction probabilities as the weight also contributes to performance improvement. Although videos from the unseen view are not available in the training process, the view classifiers are still able to be used to predict probabilities of the given test video belonging to each seen view, which are useful to obtain the final prediction scores.

4.4 Component analysis

To study the performance gain of different modules in our proposed DA-Net, we report the results of three variants of our DA-Net. In particular, in the first variant, we remove the view-prediction-guided fusion module, and only keep the basic multi-branch module and message passing module, which is referred to as *DA-Net (w/o fus.)*. Similarly in the second variant, we remove the message passing module, and only keep the basic multi-branch module and view-prediction-guided fusion module, which is referred to as *DA-Net (w/o msg.)*. In the third variant, we only keep the basic multi-branch module, which is referred to as *DA-Net (w/o msg. and fus.)*. Specially in *DA-Net (w/o msg. and fus.)* and *DA-Net (w/o fus.)*, since the fusion part is ablated, we only train one classifier for each branch, and we equally fuse the prediction scores from all branches for obtaining the action recognition results.

We take the NTU dataset under the cross-view setting as an example for component analysis. The baseline TSN method [35] is also included for comparison. Moreover, we further report the results from an ensemble version of TSN, in which we train two TSN's based on the videos from view 2 and the videos from view 3 individually, and then average their prediction scores on the test videos from view 1 for prediction results. We refer to it as *Ensemble TSN*.

The results of all methods are shown in Table 4. We observe that both Ensemble TSN and our *DA-Net (w/o msg. and fus.)* achieve better results than

Table 4. Accuracy for cross-view setting on the NTU dataset. The second and third columns are the accuracies from the RGB-stream and flow-stream, respectively. The final results after fusing the scores from the two streams are shown in the fourth column.

Method	RGB-stream	Flow-stream	Two-stream
TSN [35]	66.5%	82.2%	85.4%
Ensemble TSN	69.4%	86.6%	87.8%
DA-Net (w/o msg. and fus.)	73.9%	87.7%	89.8%
DA-Net (w/o msg.)	74.1%	88.4%	90.7%
DA-Net (w/o fus.)	74.5%	88.6%	90.9%
DA-Net	75.3%	88.9%	**92.0%**

the baseline TSN method, which indicates that learning individual representation for each view helps to capture view-specific information, and thus improves the action recognition accuracy. Our *DA-Net (w/o msg. and fus.)* outperforms the Ensemble TSN method for both modalities and after two-stream fusion, which indicates that learning common features (*i.e.* view-independent features) shared by all branches for *DA-Net (w/o msg. and fus.)* will possibly lead to better performance.

Moreover, by additionally using the message passing module, *DA-Net (w/o fus.)* gains consistent improvement over *DA-Net (w/o msg. and fus.)*. A possible reason is that videos from different views share complementary information, and the message passing process could help refine the feature representation on each branch. The *DA-Net (w/o msg.)* is also better than *DA-Net (w/o msg. and fus.)*, which demonstrates the effectiveness of our view-prediction-guided fusion module. Our DA-Net effectively integrate the predictions from all view-specific classifiers in a soft ensemble manner. In the view-prediction-guided fusion module, all the view-specific classifiers integrate the total $V \times V$ types of cross-view information. Meanwhile, the view classifier softly ensembles the action prediction scores by using view prediction probabilities as the weights.

5 Conclusion

In this paper, we have proposed the Dividing and Aggregating Network (DA-Net) to address action recognition using multi-view videos. The comprehensive experiments have demonstrated that our newly proposed deep learning method outperforms the baseline methods for multi-view action recognition. Through the component analysis, we demonstrate that view-specific representations from different branches can help each other in an effective way by conducting message passing among them. It is also demonstrated that it is beneficial to fuse the prediction scores from multiple classifiers by using the view prediction probabilities as the weights.

Acknowledgement This work is supported by SenseTime Group Limited.

References

1. Baradel, F., Wolf, C., Mille, J.: Human action recognition: Pose-based attention draws focus to hands. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2017)
2. Baradel, F., Wolf, C., Mille, J.: Pose-conditioned spatio-temporal attention for human action recognition. arXiv preprint arXiv:1703.10106 (2017)
3. Chen, W., Xiong, C., Xu, R., Corso, J.J.: Actionness ranking with lattice conditional ordinal random fields. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 748–755 (2014)
4. Chu, X., Ouyang, W., Li, H., Wang, X.: Structured feature learning for pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4715–4723 (2016)
5. Chu, X., Ouyang, W., Wang, X., et al.: Crf-cnn: Modeling structured information in human pose estimation. In: Advances in Neural Information Processing Systems. pp. 316–324 (2016)
6. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1933–1941 (2016)
7. Gorban, A., Idrees, H., Jiang, Y.G., Roshan Zamir, A., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/ (2015)
8. Gupta, A., Martinez, J., Little, J.J., Woodham, R.J.: 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2601–2608 (2014)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
10. Kong, Y., Ding, Z., Li, J., Fu, Y.: Deeply learned view-invariant features for cross-view action recognition. IEEE Transactions on Image Processing **26**(6), 3028–3037 (2017)
11. Li, R., Zickler, T.: Discriminative virtual views for cross-view action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2855–2862. IEEE (2012)
12. Li, W., Xu, Z., Xu, D., Dai, D., Van Gool, L.: Domain generalization and adaptation using low rank exemplar svms. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
13. Luvizon, D.C., Picard, D., Tabia, H.: 2d/3d pose estimation and action recognition using multitask deep learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
14. Mancini, M., Porzi, L., Rota Bul, S., Caputo, B., Ricci, E.: Boosting domain adaptation by discovering latent domains. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
15. Niu, L., Li, W., Xu, D.: Multi-view domain generalization for visual recognition. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
16. Niu, L., Li, W., Xu, D., Cai, J.: An exemplar-based multi-view domain generalization framework for visual recognition. IEEE transactions on neural networks and learning systems (2016)

17. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with fisher vectors on a compact feature set. In: Proceedings of the IEEE international conference on computer vision. pp. 1817–1824 (2013)
18. Rahmani, H., Mian, A.: Learning a non-linear knowledge transfer model for cross-view action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2458–2466 (2015)
19. Rahmani, H., Mian, A., Shah, M.: Learning a deep model for human action recognition from novel viewpoints. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
20. Ristovski, K., Radosavljevic, V., Vucetic, S., Obradovic, Z.: Continuous conditional random fields for efficient regression in large fully connected graphs. In: AAAI. pp. 840–846 (2013)
21. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1010–1019 (2016)
22. Shahroudy, A., Ng, T.T., Gong, Y., Wang, G.: Deep multimodal feature analysis for action recognition in rgb+ d videos. IEEE transactions on pattern analysis and machine intelligence (2017)
23. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1049–1058 (2016)
24. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
25. Sun, L., Jia, K., Yeung, D.Y., Shi, B.E.: Human action recognition using factorized spatio-temporal convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4597–4605 (2015)
26. Sun, S., Kuang, Z., Sheng, L., Ouyang, W., Zhang, W.: Optical flow guided feature: A fast and robust motion representation for video action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
27. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
28. Turaga, P., Veeraraghavan, A., Srivastava, A., Chellappa, R.: Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 2273–2286 (2011)
29. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional random fields for activity recognition. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. p. 235. ACM (2007)
30. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 3169–3176. IEEE (2011)
31. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision **103**(1), 60–79 (2013)
32. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)

33. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.C.: Cross-view action modeling, learning and recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2649–2656 (2014)
34. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4305–4314 (2015)
35. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: towards good practices for deep action recognition. In: European Conference on Computer Vision. pp. 20–36. Springer (2016)
36. Wang, Y., Song, J., Wang, L., Van Gool, L., Hilliges, O.: Two-stream sr-cnns for action recognition in videos. In: BMVC (2016)
37. Wu, X., Xu, D., Duan, L., Luo, J.: Action recognition using context and appearance distribution features. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 489–496. IEEE (2011)
38. Xu, D., Ouyang, W., Alameda-Pineda, X., Ricci, E., Wang, X., Sebe, N.: Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In: Advances in Neural Information Processing Systems 30. pp. 3961–3970. Curran Associates, Inc. (2017)
39. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
40. Yang, Y., Krompass, D., Tresp, V.: Tensor-train recurrent neural networks for video classification. In: International Conference on Machine Learning. pp. 3891–3900 (2017)
41. Zhang, Z., Wang, C., Xiao, B., Zhou, W., Liu, S., Shi, C.: Cross-view action recognition via a continuous virtual path. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2690–2697 (2013)
42. Zheng, J., Jiang, Z.: Learning view-invariant sparse representations for cross-view action recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3176–3183 (2013)
43. Zheng, J., Jiang, Z., Chellappa, R.: Cross-view action recognition via transferable dictionary learning. IEEE Transactions on Image Processing **25**(6), 2542–2556 (2016)
44. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1529–1537 (2015)
45. Zolfaghari, M., Oliveira, G.L., Sedaghat, N., Brox, T.: Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)