Action Recognition in Multi-view Videos

A THESIS SUBMITTED TO
THE FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGIES
OF UNIVERSITY OF SYDNEY
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY

Dongang Wang

Supervisor: Prof. Dong Xu

School of Electrical and Information Engineering
Faculty of Engineering and Information Technologies
University of Sydney

Jan 2019

Authorship Attribution Statement

The work presented in this thesis is published as [47] in the European Conference on Computer Vision (ECCV), 2018. I am the first author of this conference paper and I did all the experiments, figures, tables and almost all parts of writing. The co-authors of the published conference paper contributed to the discussion of ideas, process management, proofreading and editorial assistance.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Student Name: Dongang Wang

Signature: **Date:**

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor Name: Prof. Dong Xu

Signature: **Date:**

Action Recognition in Multi-view Videos

Dongang Wang (Email: dongang.wang@sydney.edu.au)

Supervisor: Prof. Dong Xu

School of Electrical and Information Engineering

Faculty of Engineering and Information Technologies

University of Sydney

Copyright in Relation to This Thesis

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Student Name: Dongang Wang

Signature: **Date:**

Abstract

A long-lasting goal in the field of artificial intelligence is to develop agents that can perceive and understand the rich visual world around us. With the improvement in deep learning and neural networks, many previous difficulties in the computer vision area have been resolved. For example, the accuracy in image classification has even exceeded human being in the ImageNet challenge. However, some issues are still attractive in the community, like action recognition and its application in multi-view videos.

Based on a large number of previous works in the last few years, we propose a new Dividing and Aggregating Network (DA-Net) to address the problem of action recognition in multi-view videos in this thesis. First, the DA-Net can learn view-independent representations shared by all views at lower layers and 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 are conducted accordingly. The experiments on three benchmark datasets clearly demonstrate the effectiveness of our proposed DA-Net for multi-view action recognition. We also conduct the ablation study, which indicates the three modules we proposed can provide steady improvements to the prediction accuracy.

Keywords

Convolutional Neural Network (CNN), Computer Vision, Multi-view Action Recognition, Dividing and Aggregating Network (DA-Net)

Acknowledgments

I would like to express my sincere gratefulness to my supervisor Prof. Dong Xu. He supported all my work and encouraged me to explore a lot in the area of computer vision and transfer learning. Without his selfless help, his carefulness or his rigorous guidance, I could not finish my study or publish a paper in the top conference.

Meanwhile, Dr. Wanli Ouyang also plays a crucial role in my research. He led me into the area of deep learning, taught me to use the platforms and discussed every technical detail in the thesis with me. I would also want to thank Dr. Wen Li from ETH Zürich. Dr. Li taught me how to write a successful scientific paper with every effort and patience. Besides, my teachers, colleagues, and partners from the Chinese University of Hong Kong, Shenzhen Institute of Advanced Technology and The University of Sydney all provided constructive ideas and assistance to my research. In the final stage of the work, they help a lot in accelerating the examination process. I want to thank them all.

My wife Yuting Zhang has encouraged and supported me when I was facing difficulties in researches or daily life. She has sacrificed much to help me to pursue my goals in research. I would like to thank her for everything she has done.

Thank you for this wonderful journey. I am glad that I have learned a lot.

Table of Contents

Chapter 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 [44, 39, 53, 37, 60].

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 [64, 65, 50].

1.1 Motivations

One motivation of this thesis is to learn view-specific deep representations. This is different from existing approaches for extracting view-invariant features using global codebooks [45, 32, 33] or dictionaries [65]. Because of the large divergence in specific settings of viewpoint, the visible regions are different, which 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 the 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.

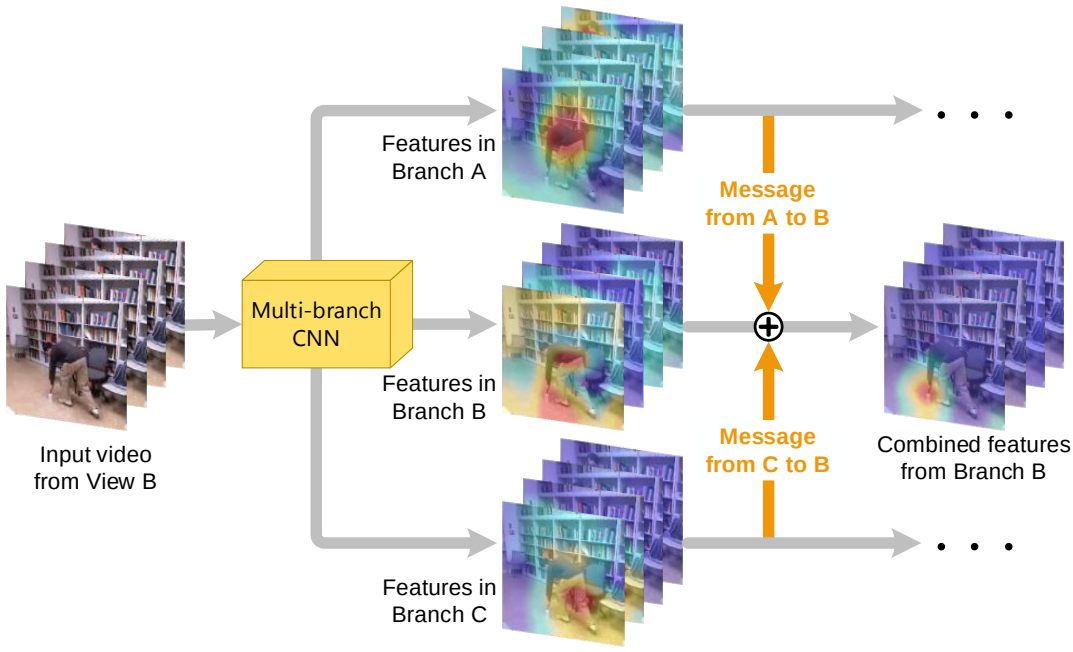


Figure 1.1: The motivation of our work for learning view-specific deep representations and passing messages among them. The features extracted in different branches should focus on different regions related to the same action. Message passing from different branches will help each other and thus improve the final classification performance. We only show the message passing from other branches to Branch B for better illustration.

Another motivation of this thesis 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 in encoding the same action. This provides us with the opportunity to pass message among these features so that they can help each other through interaction. Take Fig. 1.1 as an example, for the same input image from View B, the features from branches A, B, C focus on different regions and different angles of the same action. 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 messages 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.

1.2 Contributions

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 messages 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.

1.3 Organization of the thesis

The rest of this thesis is organized as follows. Chapter 2 introduces recent methods that are related to deep learning and action recognition, especially the methods for multi-view action recognition. Chapter 3 illustrates the definition of our newly proposed Dividing and Aggregating Network (DA-Net). The structure of our DA-Net is described as a combination of three modules. Our implementation of the DA-Net for training and testing is described in Chapter 4. The experimental results on different datasets are summarized in Chapter 5. We have conducted experiments in two settings, including the *cross-subject* setting to predict videos from different subjects and the *cross-view* setting to predict videos from unseen views. Finally, we conclude our design in Chapter 6.

Chapter 2

Literature Review

The problems related to action recognition have been studied for decades, and the techniques for action recognition could be described in three aspects. The first aspect is to treat the actions as stacks of pictures. From this point, the works in convolutional neural networks mainly for image classification could be utilized. Secondly, the video signals perform in time sequence, which enables the techniques like trajectory methods [49], recurrent neural network [12] and attention mechanism [1] in the action recognition problems. Besides, specific techniques like conditional random field (CRF) [66] can bring insights into specific multi-view action recognition problems.

For the literature review, the basic deep learning methods will be first introduced, followed by specific methods for action recognition. The methods for multi-view action recognition and usage of CRF will also be discussed afterward.

2.1 Deep Learning Structures

For this section, the structures for neural networks (*i.e.* deep learning) are summarized, including the Convolutional Neural Networks (CNN) for image classifications and the Recurrent Neural Networks (RNN) for sequence modeling problems. Both of these structures are widely used in action recognition problems.

2.1.1 Convolutional Neural Networks and Back-propagation

The early version of convolutional neural networks (CNN) was introduced in 1982 as Neocognitron [11], where the authors introduced the hierarchy model to distinguish written digits. The

idea of this paper [11] comes from the findings in the visual nervous system of the vertebrate, which consists of two kinds of cells as simple cells and complex cells that process different levels of information. However, this structure only provides a forward computing. Later in 1986, Rumelhart *et al.* [56] published a paper and proposed a computing method called back-propagation. By defining a loss function at the end of the network and by conducting chain rule, the result could be propagated back to every neuron and update the parameters. This is the mathematical background knowledge of all neural networks.

One milestone is a back-propagated convolutional neural network structure called LeNet [22] proposed by LeCun *et al.* in order to classify the written zip code MNIST dataset [21]. The structure contains five layers of filters (called 'kernels'), and the number of filters is different in different layers. The convolutional computation is conducted by traversing the filters over the output of the previous layer (called 'feature maps'). After each convolutional layer, a pooling layer performs to select the focused points in the feature map. The structure has influenced the other works in deep learning. For example, in 2012, Krizhevsky *et al.* established one powerful neural network on two GPUs and won the ImageNet Challenge [8], and the result outperformed the rest methods by a large margin. The network is called *AlexNet* [20]. The differences between *AlexNet* and *LeNet* are mainly in the network structure and optimization procedures. In *AlexNet*, overlapping max pooling was utilized instead of average pooling in *LeNet*. *AlexNet* also used ReLU as activation function instead of Sigmoid in *LeNet*. Besides, *AlexNet* contains more neurons than *LeNet*, which increases the capacity of the model.

At present, the frequently used structures in computer vision community are *VGG* [38], *Inception* [43] and *ResNet* [15] combined with different tricks, such as Dropout and Batch Normalization [17]. BN-Inception [17] serves as an example, which is similar to *GoogLeNet* [43] but did changes in the number of filters and method of pooling. In the paper of BN-Inception [17], the authors proposed an idea that when the data within the different mini-batches could be transformed into one normal distribution, the parameters learned in each neuron would be more steady and contain more semantic information. Supposing the situations that the original distribution could provide good enough output, another layer after this normalization is added to enable the network to compute reversely. The results are good for image classification and action recognition, and this network is utilized in later works like the temporal segment network (TSN) [53].

2.1.2 Recurrent Neural Networks and LSTM

Another pattern of neural networks is called recurrent neural networks (RNN), in which the data are treated as time sequences instead of time independent signals in CNN. The goal is achieved by the hidden layer in RNN, which could store the state of each time step and pass the state to the next time step.

A crucial problem has been discovered by using RNN, which is the network could only store states for a short term, and the states of the previous stages could be vanished or exaggerated after several steps. To solve this problem, an advanced version of RNN is proposed by Hochreiter *et al.* [16], which is called Long Short-Term Memory (LSTM) structure. The LSTM block exploits a more complex memory cell to store all the previous hidden states, and the forget gate, memory gate, and output gate are all learned accordingly. This method is proved to be useful in sequence modeling problems.

A common method of using LSTM in action recognition is to use CNN to extract features from raw images and the features are fed into LSTM to encode time-based information and generate the predicted class of action for the output. In [61], the authors used GoogLeNet to extract features and used stacked LSTM to conduct prediction based on the feature. To be more clarified, the stacked LSTM contains five layers, and each layer contains 512 memory cells. Following the LSTM layers, a softmax classifier makes a prediction at every input frame feature. In [9], the authors proposed a similar structure with a single-layer LSTM. They also expanded the structure to visual captioning tasks in which the output of LSTM are sequences of words forming into natural sentences. However, the performances of such structures are not as impressive as the methods based on CNNs, so we didn't use RNN-based methods for multi-view action recognition.

2.2 Methods in Action Recognition

Researchers have made significant contributions in designing effective features as well as classifiers for action recognition [29, 49, 54, 52, 42]. Wang *et al.* [48] proposed the improved Dense Trajectory (iDT) feature to encode the information from the edge, flow and trajectory. The iDT feature became dominant in the THUMOS 2015 Challenge [13]. This method is an expansion of optical flow in which the descriptors of each frame are counted and combined together to

form into a large feature. HOF, HOG and MBH descriptors are utilized, and the final length of one trajectory is 436. One video will contain many trajectories and these trajectory features are used to train a support vector machine for each action.

In the deep learning community, Tran *et al.* proposed C3D [44], which designs a 3D CNN model for video datasets by combining appearance features with motion information. Sun *et al.* [41] 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 [39] first proposed the two-stream CNN to extract features from the RGB keyframes and the optical flow channels. Wang *et al.* [52] integrated the key factors from iDT and CNN and achieved significant performance improvement. Wang *et al.* also proposed the temporal segment network (TSN) [53] to utilize segments of videos under the two-stream CNN framework. The TSN network reported the state-of-the-art results on UCF101 dataset [40] with the accuracy of around 95%. In this work, the authors proposed a two-stream CNN network, which takes RGB images as inputs for one stream and optical flow images for the other stream. The two CNN network both use BN-Inception [17] as the backbone, and the final scores of each video are the fusion of the results from two streams. Small but effective tricks are use in TSN. For example, to utilize the models that are pre-trained using RGB images from ImageNet [8] to optical flow images, the authors resampled the optical flow images to 256-level grayscale images and merged the three color channels of the pre-trained model to one channel to match the grayscale images. Our network uses TSN as the baseline and uses the corresponding tricks.

Researchers also transform the two-stream structure to the multi-branch structure. In [10], 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 levels of features and then fuse them together [54]. These works define multi-branch structures to deal with different modalities of videos instead of videos from different viewpoints. Therefore, they do not learn view-specific features for multi-view videos or use the prior to fuse the classification scores from multiple branches as in our work. We use the multi-branch structure in order to deal with the videos from different viewpoints, and the two-stream structure is conducted at the same time to handle the two common modalities, *i.e.* RGB and optical flow.

2.3 Methods related to Multi-view Action Recognition

2.3.1 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 [64, 50, 27, 28]. The methods using view-invariant representations are popular for multi-view action recognition. Wu *et al.* [57] and Turaga *et al.* [45] 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.* [65], Kong *et al.* [19] and Hossein *et al.* [33] 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.* [24] and Mancini *et al.* [26] 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 propose to 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.

2.3.2 Conditional Random Field (CRF)

CRF has been exploited for action recognition in [46] as it can connect features and outputs, especially for temporal signals like actions. Chen *et al.* proposed L-CORF [5] 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.* [66], which deals with the relationship among pixels. Xu *et al.* [59, 58] modeled the relationship of pixels to learn the edges of objects in images. Recently, Chu *et al.* [6, 7] have utilized discrete CRF in CNN for human pose estimation.

Different from the previous applications using CRF, our work is the first to use CRF 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.

2.4 Summary and Discussion

The basic ideas of convolutional neural networks and recurrent neural networks are first introduced, which are the mainstream methods in nowadays action recognition. Some specific methods for action recognition are reviewed, including methods based on iDT and two-stream CNNs. As for multi-view action recognition, the previous works are reviewed. Specifically, the previous applications of CRF are introduced, and to the best of my knowledge, it was not previously used in multi-view action recognition problems.

By conducting comparisons between the traditional methods (*e.g.* iDT) and the deep learning methods (*e.g.* TSN), we could find some similarities and dissimilarities in dealing with videos and action recognition problems. The optical flow is a powerful feature, for it can encode the spatial and temporal information at the same time. In that case, the two-stream networks utilize the optical flow feature to build a separate stream, and we use the widely used two-stream network TSN [53] as our backbone. Besides, researchers have used ideas from the traditional methods in the neural networks. For example, when extracting optical flow features from frames in the work of Wang *et al.* [48], the camera motions and human motions are detected to fine-grain optical flow in order to indicate better real motions. This technique is used in TSN [53] to define the wrapped optical flow. Our usage of CRF also follows this philosophy by moving the method from the graphical models to neural networks for better performances.

Chapter 3

Dividing and Aggregating Network (DA-Net) for 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. 3.1, this network consists of three modules. (1) **Basic Multi-branch Module**: This network extracts the common 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

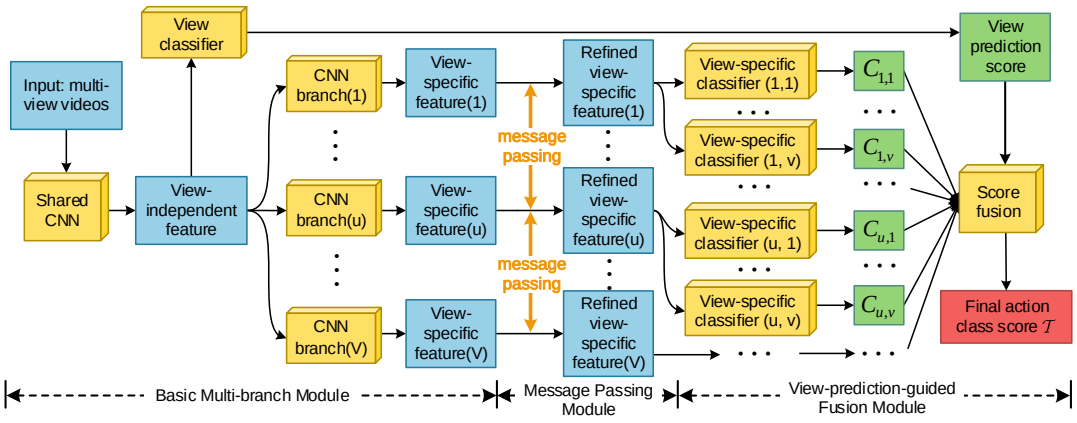


Figure 3.1: Network structure of our newly proposed Dividing and Aggregating Network (DA-Net). (1) **Basic multi-branch module** is composed of one shared CNN and several view-specific CNN branches. (2) **Message passing module** is introduced between every two branches and generate the refined view-specific 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.

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. 3.1, 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 effectively 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. 3.2(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,

$$E(\mathbf{H}, \mathbf{F}, \Theta) = \sum_{v} \phi(\mathbf{h}_v, \mathbf{f}_v) + \sum_{u,v} \psi(\mathbf{h}_u, \mathbf{h}_v), \tag{3.1}$$

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,

$$\phi(\mathbf{h}_v, \mathbf{f}_v) = -\frac{\alpha_v}{2}\|\mathbf{h}_v - \mathbf{f}_v\|^2, \tag{3.2}$$

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

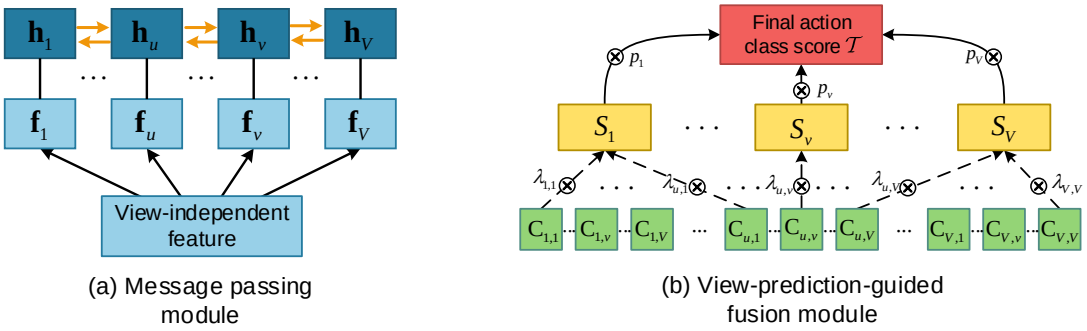
$$\psi(\mathbf{h}_u, \mathbf{h}_v) = \mathbf{h}_v{}^{\top}\mathbf{W}_{u,v}\mathbf{h}_u, \tag{3.3}$$

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

$$\mathbf{h}_v = \frac{1}{\alpha_v}(\alpha_v\mathbf{f}_v + \sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u)). \tag{3.4}$$

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



(a) Message passing module

(b) View-prediction-guided fusion module

Figure 3.2: 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.

From the definition of CRF, the first term in Eqn.(3.4) 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.(3.3) and Eqn.(3.4) 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 [66, 7], thus it can be naturally integrated with the basic multi-branch network, and optimized based on the basic multi-branch 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.

3.4.1 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. 3.2(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^i_v can be formulated as follows,

$$S^i_v = \sum_u \lambda_{u,v} C^i_{u,v}, \tag{3.5}$$

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 ($u \neq v$).

3.4.2 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,

$$\mathcal{T}^i = \sum_{v=1}^{V} p_v^i S_v^i. \tag{3.6}$$

To obtain the view prediction probabilities, as shown in Fig. 3.1, 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.*,

$$\mathcal{L} = \mathcal{L}_{action} + \mathcal{L}_{view}, \tag{3.7}$$

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 5.3).

Chapter 4

Using DA-Net for Training and Testing

4.1 Network Architecture

We illustrate the architecture of our DA-Net in Fig. 3.1. The shared CNN can be any of the popular CNN architectures, which is followed by 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.(3.5). 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.

We build our network based on the temporal segment network(TSN) [53] with some modifications. In particular, we use the BN-Inception [17] as the backbone network for experiments. The shared CNN layers include the ones from the input to the block `inception_5a`. As shown in Fig. 4.1, for each path within the `inception_5b` block, we duplicate the last convolutional layer (shown in red in Fig. 4.1) 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 the initialization stage and learnt separately (*i.e.*, the weights in the branches are not shared). Similarly as in TSN, we also train a two-stream network [39], where two streams are learnt separately using two modalities, RGB (referred to as the **RGB-stream**) and dense optical flow (referred to as the **Flow-stream**), respectively. In the testing phase, given a test sample with multiple views of videos, $(\mathbf{x}_1, \ldots, \mathbf{x}_V)$, we pass each

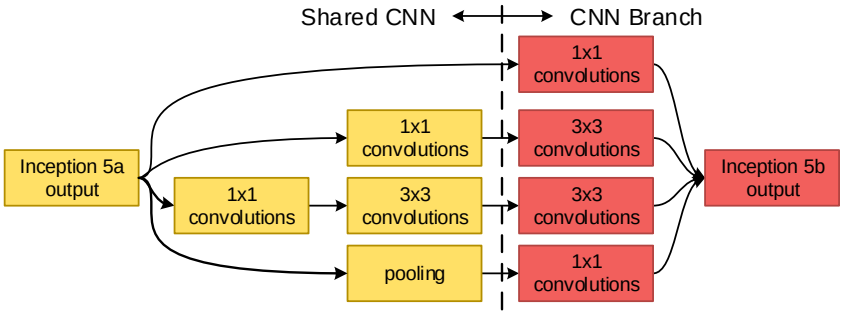


Figure 4.1: The layers used in the shared CNN and CNN branches in the `inception_5b` block. The layers in yellow color are included in the shared CNN, while the layers in red color are duplicated for different branches. The layers after `inception_5b` are also duplicated. The `ReLU` and `BatchNormalization` layers after each convolutional layer are treated similarly as the corresponding convolutional layers.

video $\mathbf{x}_v$ to two streams and obtain its prediction by fusing the outputs from two streams.

4.2 Training Details

Like other deep neural networks, our proposed model can be trained by using popular optimization approaches such as 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 by additionally adding the message passing module and view-prediction-guided fusion module. Without using this two-step approach (*i.e.* we learn the whole network in one step), the accuracy will drop because the network starts to pass messages before the branches are ready to encode view-specific features.

The training of our DA-Net has the same starting point of TSN in order to keep consistency with TSN and other works. The initialization is the same as the steps in TSN. We use the parameters of BN-Inception [17] pre-trained on ImageNet [8] as the initialization for the RGB-stream. For the Flow-stream, we follow the cross modality pre-training technique introduced in TSN [53], where we average the weights of the first convolutional layer across the three channels in RGB-stream and duplicate the averaged weights by the number of optical flow channels (which is 10 in our work). Following TSN [53], we also use the TVL1 algorithm [62] to extract dense optical flow. The input to the Flow-stream contains 10 channels, including 5 consecutive grayscale optical flow images in x-direction and 5 grayscale optical flow images of the same time in y-direction.

Our network is built on Caffe [18] and can be trained on one NVIDIA GeForce GTX1080 Ti graphical card. The batch size is 32 for both RGB-stream and Flow-stream in the training stage of the basic multi-branch module and the fine-tuning stage of the whole DA-Net. For the datasets with smaller sizes (like the dataset NUMA [51] and IXMAS [55] in Chapter 5), the base learning rate is set as 0.001 for both streams, which will be divided by 10 after every 30 epochs, and the total epoch for training is 100. For the datasets with larger sizes (like the dataset NTU [35] in Chapter 5), we use smaller base learning rate as 0.0001 and smaller total epoch as 50 for both streams, and the learning rate will also be divided by 10 after every 16 epochs.

Like in TSN, the input to the networks are segments of videos. We use three segments for videos by default. For videos that are very short (*e.g.* some videos in the dataset NUMA [51]), we select the segments with overlaps. For the rest settings, we use the default values. We use 0.9 for the rate of momentum, and 0.0005 for weight decay. The network may suffer the explosion in gradient values, so we use the clip gradient mechanism in Caffe [18]. We set the upper bound of the gradients as 20 and 40 for Flow-stream and RGB-stream respectively, which are the same setting as TSN [53].

4.3 Testing Details

Our testing stage also follows the steps of TSN [53]. For each video, 25 frames are evenly extracted from the video and fed into the RGB-stream and 25 flow stacks are fed into the Flow-stream. The scores are computed according to the 25 images for each stream and the final scores are combined by using a manually defined rate. We use the default combination rate from TSN [53], which are 1 and 1.5 for results from RGB-stream and Flow-stream respectively.

When dealing with videos that are too short that contain fewer frames than 25 (*e.g.* some videos in the dataset NUMA [51]), the total numbers of frames taken for testing are different. We use 8 frames for both RGB-stream and Flow-stream in our experiments, which will provide acceptable performances.

Since we define and train a view classifier for videos from multiple viewpoints in the training stage, the view labels are not needed for testing. Instead, the videos will go through every branch and the view classifier will generate the view prediction scores for each video, which are used for the fusion of the action recognition results from all branches.

Chapter 5

Experiments on DA-Net

In this chapter, we conduct experiments to evaluate our proposed model by using three 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.

5.1 Datasets and Setup

NTU RGB+D (NTU) [35] 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)[51] is another popular multi-view action recognition benchmark dataset. In this dataset, 10 daily actions[1] 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.

[1]The 10 actions are *pick up with one hand*, *pick up with two hands*, *drop trash*, *walk around*, *sit down*, *stand up*, *donning*, *doffing*, *throw*, and *carry*.

IXMAS [55] is a widely used multi-view action recognition dataset. Following the experimental setting in the existing works [55, 45], we conduct the experiments by using 11 daily actions performed by 10 subjects[2]. Each action is performed 3 times (each time of each action is referred to as one *trial*) by each person with different orientations, which leads to in total 330 *trials*. Each *trial* is recorded by 5 different cameras from different viewpoints, so the total number of videos from all viewpoints is $1,650$.

According to the previous works on multi-view action recognition [55, 45, 51, 35], the released versions of these datasets contain multiple modalities, such as RGB frames, binary silhouette images (in IXMAS only) and skeleton coordinates (in NUMA and NTU). We only utilize the RGB frames without knowing the ground-truth background images in our experiments. Since the optical flow is extracted from the original RGB images, we only use the RGB images compared with other works (See Table 5.1).

5.2 Experiments on Multi-view Action Recognition

The *cross-subject* evaluation protocol is used in this section. 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[3] as in [35]. We compare our proposed method with a wide range of baselines, among which the work in [35, 36, 2] include 3D joint information, and the work in [3, 25] used RGB videos only. We also include the TSN method [53] 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 5.1. 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.* [3] and Luvizon *et al.* [25] reported the results using only RGB videos, where the work from Luvizon *et al.* [25] achieves similar performance as the TSN method. Our proposed DA-Net outperforms all existing state-of-the-art algorithms and

[2]The 11 daily action classes are *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, and *pick up*.

[3]The subject IDs in the training set are 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38 and the remaining subjects are reserved for testing.

Table 5.1: 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 [53] and the work from Zolfaghari *et al.* [67] 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 [36]	Pose+RGB	74.9%	-
STA-Hands [2]	Pose+RGB	82.5%	88.6%
Zolfaghari *et al.* [67]	Pose+RGB	80.8%	-
Baradel *et al.* [3]	Pose+RGB	84.8%	90.6%
Luvizon *et al.* [25]	RGB	84.6%	-
TSN [53]	RGB	84.93%	85.36%
DA-Net (Ours)	RGB	**88.12%**	**91.96%**

the baseline TSN method.

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 5.2, where our proposed DA-Net again outperforms all other baseline methods.

For the IXMAS dataset, we adopt the same leave-one-subject-out training scheme as in [45, 55]. For each time of training, all the videos of one same subject are treated as the test set, and all the rest videos from the other subjects are used as the training set. To keep the consistency with previous works, the final results are generated by fusing scores from all synchronized five views for each trial. We averagely fuse all the five video prediction scores for one trial. Considering all ten actors acting each of the eleven actions for three times, the total number of trials should be 330 ($10 \times 11 \times 3$), and the accuracy is the total correctly-predicted trial number divided by the total number of trials. We report the results and compare them with the corresponding state-of-the-art works in Table 5.3.

According to Table 5.3, our network achieves better performance than the previous methods as well as the baseline TSN itself although the dataset is almost saturated. For trial-level performance, only three out of 330 instances are wrongly predicted. Two incorrect videos from 'Check Watch' are predicted as 'Punch' because the body movements in the videos are

Table 5.2: 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 [23]	50.7%
MST-AOG [51]	81.6%
Kong *et al.* [19]	81.1%
TSN [53]	90.3%
DA-Net (ours)	**92.1%**

Table 5.3: Accuracy (%) comparison between our DA-Net and other works on the IXMAS dataset. The numbers in brackets indicate the way how accuracy is computed, by computing the proportion of correctly-predicted trial number and the total number of trials. The total trial number is 330, and only three of 330 are predicted wrongly in our DA-Net.

Method	Accuracy
Weinland *et al.* [55]	93.33 (308/330)
Turaga *et al.* [45]	98.78 (326/330)
Wu *et al.* [57]	90.6 (299/330)
Burghouts *et al.* [4]	96.4 (318/330)
TSN [53]	98.48 (325/330)
DA-Net (ours)	**99.09** (327/330)

more intense compared with other 'Check Watch' actions. One video from 'Scratch Head' is predicted as 'Wave' because the video stops once the hand reaches the head so that less information could be figured out. For video-level performance when considering the videos from different views separately, the baseline TSN could reach accuracy to 95.7% and DA-Net outperforms it by decreasing error rate by around 30%, to the accuracy of 97.0%.

The results on these 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.

Table 5.4: Average accuracy comparison on the NUMA dataset [51] (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 the 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 [63]	58.5%	55.2%	39.3%	51.0%
nCTE [14]	68.6%	68.3%	52.1%	63.0%
MST-AOG [51]	-	-	-	73.3%
NKTM [32]	75.8%	73.3%	59.1%	69.4%
R-NKTM [33]	78.1%	-	-	-
Kong *et al.* [19]	-	-	-	77.2%
TSN [53]	84.5%	80.6%	76.8%	80.6%
DA-Net (ours)	**86.5%**	**82.7%**	**83.1%**	**84.2%**

5.3 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.

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 [35], 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 5.1. 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

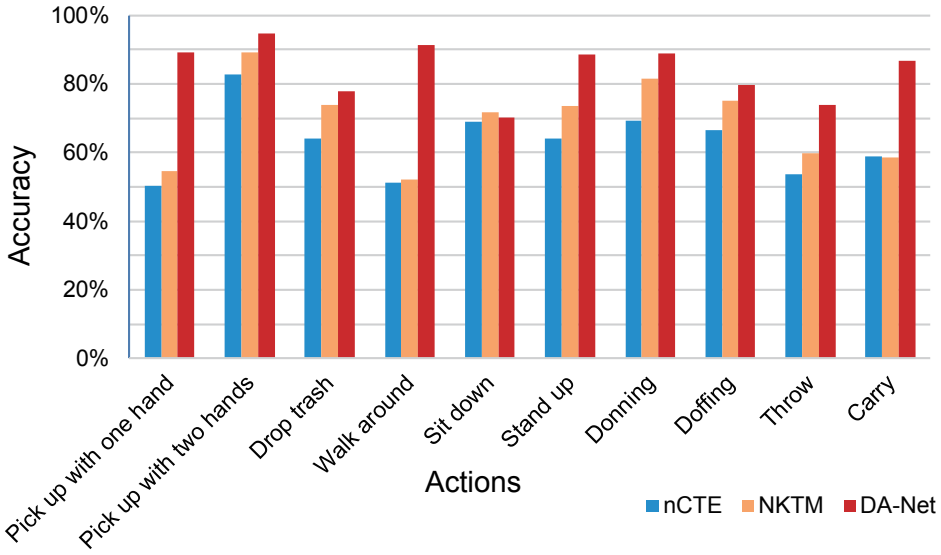


Figure 5.1: Average recognition accuracy in each class on the NUMA dataset under the cross-view setting. All the three methods do not utilize the features from the unseen view during the training process.

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 5.4, which shows our DA-Net achieves the best performance compared with other works. Our results are even better than the methods that use the videos from the unseen view as unlabeled data in [19]. The detailed accuracy for each class is shown in Fig. 5.1. Again we observe that DA-Net is better than nCTE [14] and NKTM [32] 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 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 classifier is still able to be used to predict probabilities of the given test video resembling each seen view, which are useful to obtain the final prediction scores.

Table 5.5: 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 [53]	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%**

5.4 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 [53] 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 5.5. We observe that both Ensemble TSN and our *DA-Net (w/o msg. and fus.)* achieve better results than 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.

5.5 Visualization

We use the toolbox DeepDraw [30] to visualize our DA-Net model and compare it with the TSN [53] model. We use the model from the RGB-stream to conduct visualization, as it contains more visual semantics. The following pages are the visualization results of the classes in the NTU dataset [35] and the NUMA dataset [51].

By comparing the visualization results from TSN and our proposed DA-Net, we have the following observations.

First, our DA-Net performs better than TSN in capturing visual cues of meaningful parts and actions as shown in Fig. 5.2. For example, in the class '*tear up paper*' in the NTU dataset, the action of hands is highlighted in our approach while TSN does not capture this visual cue. We have similar observations for the classes of '*walking towards each other*' in the NTU dataset, and the classes of '*pick up with one hand*' and '*carry*' in the NUMA dataset.

Second, our DA-Net is able to generate representations from more diverse viewpoints for better descriptions of multi-view visual cues, which finally lead to better results. For example, DA-Net captures actions with more diverse viewpoints than TSN for the actions of '*sitting down*', '*sneeze/cough*', '*touch back (backache)*' and '*walking apart from each other*' in Fig. 5.3.

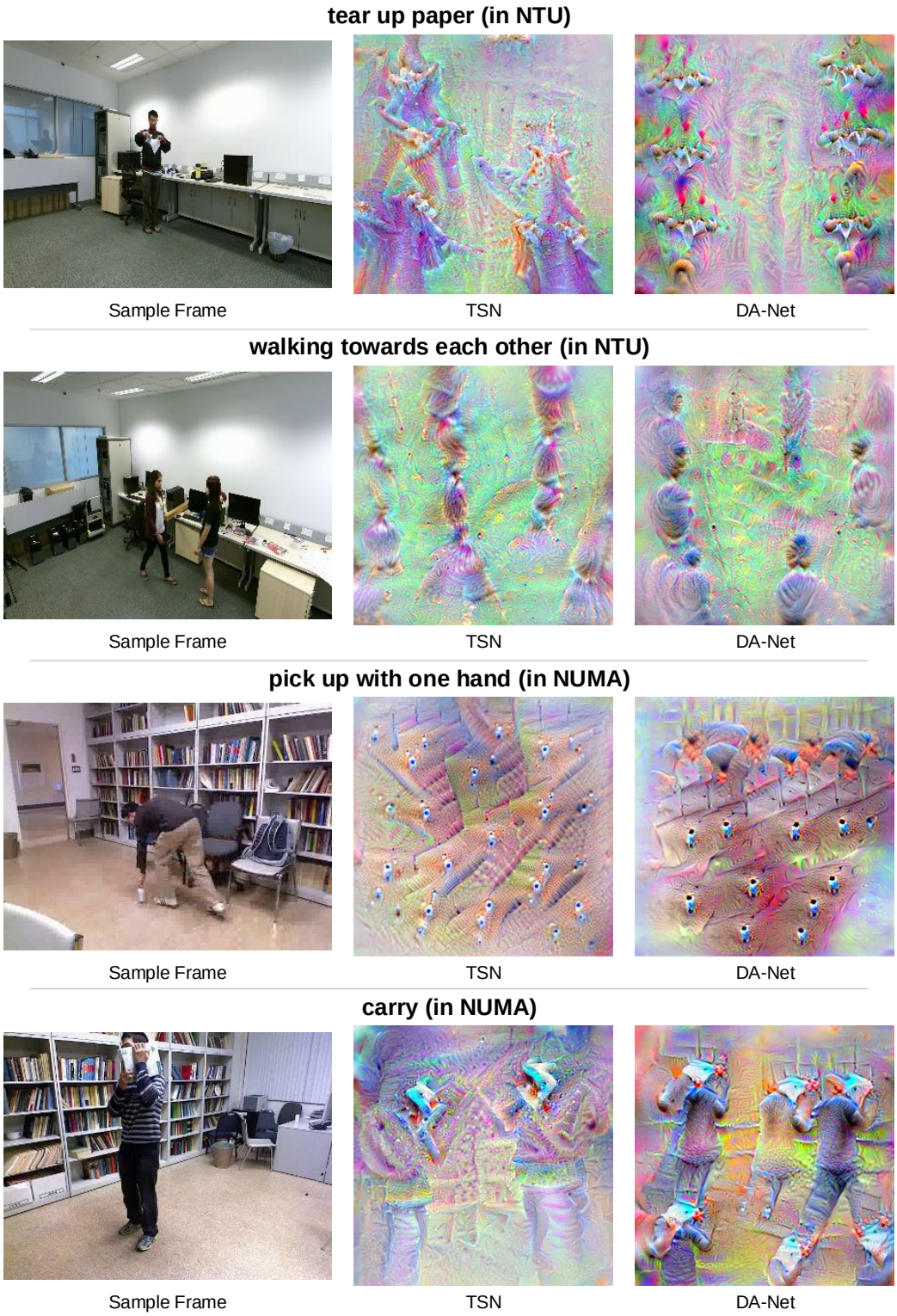


Figure 5.2: Visualization results of different actions in the datasets. For '*tear up paper*' in the NTU dataset, our DA-Net can capture the details in hands. For '*walking towards each other*' in the NTU dataset, our DA-Net can better represent the relationship of people, who are facing to the center. For '*pick up with one hand*' in the NUMA dataset, our DA-Net can capture the movement of human body instead of just focusing on the bottle to be picked up as in TSN. For '*carry*' in the NUMA dataset, our DA-Net can enhance the key information of the carried stuff.

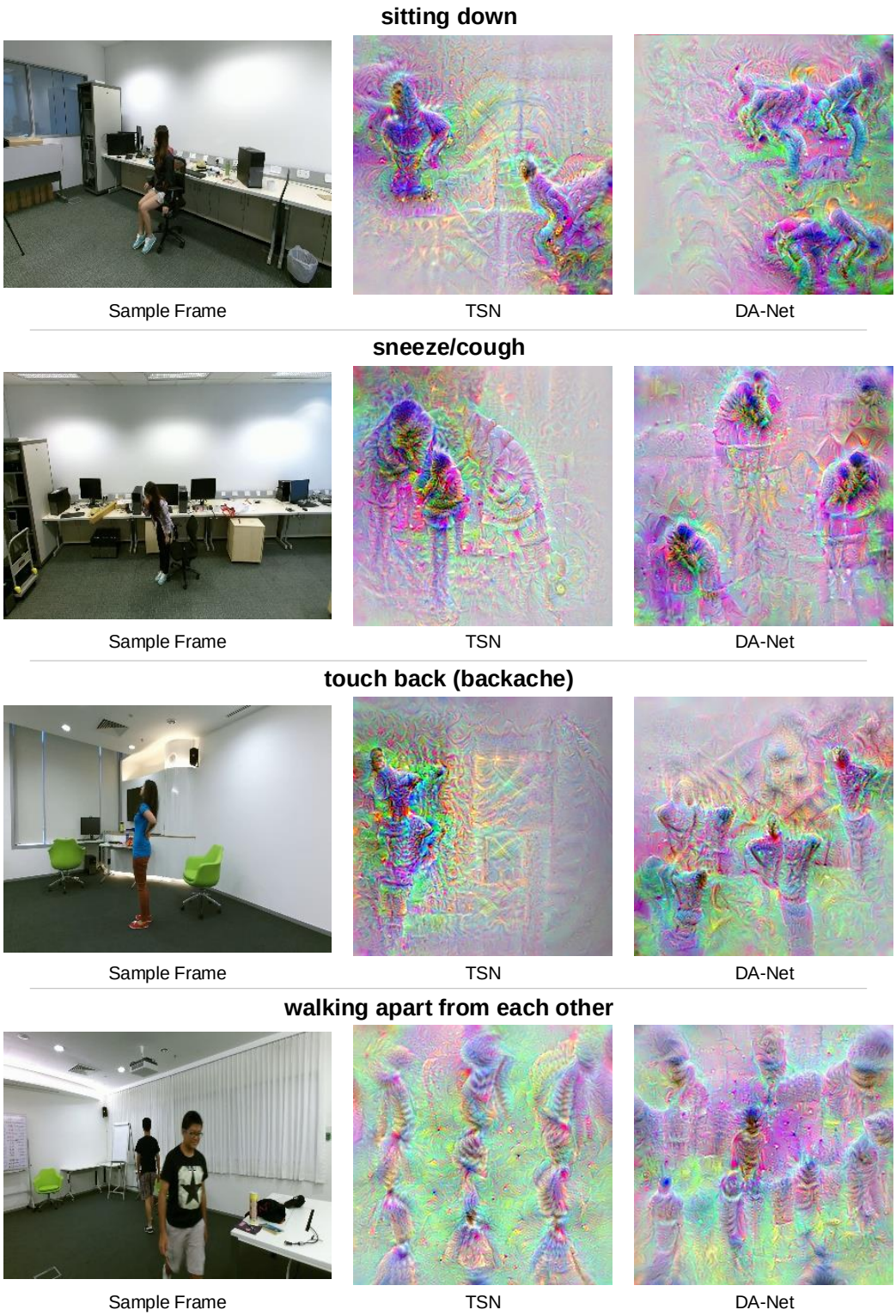


Figure 5.3: Visualization results in the NTU dataset. In these four classes, our DA-Net better integrates information from different viewpoints.

Chapter 6

Conclusions

In this work, we have proposed the Dividing and Aggregating Network (DA-Net) to address action recognition in multi-view videos. The network contains three modules. The basic multi-branch module is able to learn view-independent representations and view-specific representations. The message passing module between every two branches is used to integrate different view-specific representations and generate the refined features. We also use the view-prediction-guided fusion module to fuse the prediction results from all view-specific classifiers.

The comprehensive experiments have demonstrated that the newly proposed deep learning method DA-Net 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.

Appendix A

Details on CRF

First we define a continuous conditional random field (CRF) to model the conditional distribution of the original view-specific feature $\mathbf{F} = \{\mathbf{f}_v\}_{v=1}^V$ and the refined view-specific feature $\mathbf{H} = \{\mathbf{h}_v\}_{v=1}^V$ [31].

$$P(\mathbf{H}|\mathbf{F}, \Theta) = \frac{1}{Z(\mathbf{F})} \exp\{E(\mathbf{H}, \mathbf{F}, \Theta)\}, \tag{a1}$$

where $Z(\mathbf{F}) = \int_{\mathbf{H}} \exp\{E(\mathbf{H}, \mathbf{F}, \Theta)\} d\mathbf{H}$ is the partition function for normalization, and Θ is the set of parameters. $E(\mathbf{H}, \mathbf{F}, \Theta)$ is the energy function which is defined as

$$E(\mathbf{H}, \mathbf{F}, \Theta) = \sum_v \phi(\mathbf{h}_v, \mathbf{f}_v) + \sum_{u,v} \psi(\mathbf{h}_u, \mathbf{h}_v), \tag{a2}$$

where ϕ is the unary potential and ψ is the pairwise potential. As defined in Chapter 3,

$$\phi(\mathbf{h}_v, \mathbf{f}_v) = -\frac{\alpha_v}{2} \|\mathbf{h}_v - \mathbf{f}_v\|^2, \tag{a3}$$

$$\psi(\mathbf{h}_u, \mathbf{h}_v) = \mathbf{h}_v{}^\top \mathbf{W}_{u,v} \mathbf{h}_u. \tag{a4}$$

This is a typical formulation of CRF, which could be solved by using mean-field inference. Under the mean-field theory, the approximation of $P(\mathbf{H}|\mathbf{F})$ can be $Q(\mathbf{H}|\mathbf{F}) = \prod_{v=1}^V Q_v(\mathbf{h}_v|\mathbf{F})$ which minimizes Kullback-Leibler (KL) divergence between P and Q, and can be written as below [34],

$$\log Q_v(\mathbf{h}_v|\mathbf{F}) = \mathbb{E}_{u \neq v} \left(\log P(\mathbf{H}|\mathbf{F})\right) + const. \tag{a5}$$

The $\log Q_v(\mathbf{h}_v|\mathbf{F})$ in (a5) could be written as follows when $P(\mathbf{H}|\mathbf{F})$ is replaced by the terms in (a2)-(a4):

$$\log Q_v(\mathbf{h}_v|\mathbf{F}) = -\frac{\alpha_v}{2}\|\mathbf{h}_v - \mathbf{f}_v\|^2 + \mathbf{h}_v^\top \sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u) + const. \tag{a6}$$

After we rearrange the expression above into an exponential form, use the expansion form of the unary term and omit the constant terms, the distribution $Q_v(\mathbf{h}_v|\mathbf{F})$ could be derived into

$$Q_v(\mathbf{h}_v|\mathbf{F}) \propto \exp(-\frac{\alpha_v}{2}(\|\mathbf{h}_v\|^2 - 2\mathbf{h}_v^\top \mathbf{f}_v) + \mathbf{h}_v^\top \sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u)). \tag{a7}$$

The above formulation could be rewritten as below,

$$\begin{aligned} Q_v(\mathbf{h}_v|\mathbf{F}) &\propto \exp\left\{-\frac{\alpha_v}{2}\left(\|\mathbf{h}_v\|^2 - 2\mathbf{h}_v^\top\left(\mathbf{f}_v + \frac{1}{\alpha_v}\sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u)\right)\right)\right\}, \\ &\propto \exp\left\{-\frac{\alpha_v}{2}\left\|\mathbf{h}_v - \left(\mathbf{f}_v + \frac{1}{\alpha_v}\sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u\right)\right\|^2\right\}, \end{aligned} \tag{a8}$$

which indicates that the posterior distribution of $\mathbf{h}_v$ follows a Gaussian distribution, and its mean vector could be written as:

$$\mathbf{h}_v = \frac{1}{\alpha_v}(\alpha_v \mathbf{f}_v + \sum_{u \neq v}(\mathbf{W}_{u,v}\mathbf{h}_u)). \tag{a9}$$

Thus, the refined view-specific feature representation $\{\mathbf{h}_v|_{v=1}^V\}$ can be obtained by iteratively applying the above equation. The result is the same as Eqn.3.4 in Chapter 3.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] F. Baradel, C. Wolf, and J. Mille. Human action recognition: Pose-based attention draws focus to hands. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[3] F. Baradel, C. Wolf, and J. Mille. Pose-conditioned spatio-temporal attention for human action recognition. *arXiv preprint arXiv:1703.10106*, 2017.

[4] G. Burghouts, P. Eendebak, H. Bouma, and J.-M. ten Hove. Improved action recognition by combining multiple 2d views in the bag-of-words model. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 250–255. IEEE, 2013.

[5] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 748–755, 2014.

[6] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4715–4723, 2016.

[7] X. Chu, W. Ouyang, X. Wang, et al. Crf-cnn: Modeling structured information in human pose estimation. In *Advances in Neural Information Processing Systems*, pages 316–324, 2016.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale

hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[9] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[11] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[13] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[14] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham. 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*, pages 2601–2608, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[19] Y. Kong, Z. Ding, J. Li, and Y. Fu. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 26(6):3028–3037, 2017.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Y. LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2855–2862. IEEE, 2012.

[24] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[25] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] M. Mancini, L. Porzi, S. Rota Bul, B. Caputo, and E. Ricci. Boosting domain adaptation by discovering latent domains. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] L. Niu, W. Li, and D. Xu. Multi-view domain generalization for visual recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[28] L. Niu, W. Li, D. Xu, and J. Cai. An exemplar-based multi-view domain generalization framework for visual recognition. *IEEE transactions on neural networks and learning systems*, 2016.

[29] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*, pages 1817–1824, 2013.

[30] A. M. Øygard. Deep draw. https://github.com/auduno/deepdraw, 2015.

[31] T. Qin, T.-y. Liu, X.-d. Zhang, D.-s. Wang, and H. Li. Global ranking using continuous conditional random fields. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1281–1288. Curran Associates, Inc., 2009.

[32] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2458–2466, 2015.

[33] H. Rahmani, A. Mian, and M. Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[34] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*, pages 840–846, 2013.

[35] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019, 2016.

[36] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[37] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

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

[39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[40] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[41] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.

[42] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. 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.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[45] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. 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.

[46] D. L. Vail, M. M. Veloso, and J. D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 235. ACM, 2007.

[47] D. Wang, W. Ouyang, W. Li, and D. Xu. Dividing and aggregating network for multi-view action recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[48] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[49] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[50] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[51] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[52] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[53] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[54] Y. Wang, J. Song, L. Wang, L. Van Gool, and O. Hilliges. Two-stream sr-cnns for action recognition in videos. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 108.1–108.12. BMVA Press, September 2016.

[55] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer vision and image understanding*, 104(2):249–257, 2006.

[56] D. Williams and G. Hinton. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

[57] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 489–496. IEEE, 2011.

[58] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction.

In *Advances in Neural Information Processing Systems 30*, pages 3961–3970. Curran Associates, Inc., 2017.

[59] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. 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.

[60] Y. Yang, D. Krompass, and V. Tresp. Tensor-train recurrent neural networks for video classification. In *International Conference on Machine Learning*, pages 3891–3900, 2017.

[61] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[62] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

[63] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-view action recognition via a continuous virtual path. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2697, 2013.

[64] J. Zheng and Z. Jiang. Learning view-invariant sparse representations for cross-view action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3176–3183, 2013.

[65] J. Zheng, Z. Jiang, and R. Chellappa. Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing*, 25(6):2542–2556, 2016.

[66] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[67] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox. 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.