Journal of Systems Architecture 103 (2020) 101695

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Multi-teacher knowledge distillation for compressed video action recognition based on deep learning

Meng-Chieh Wu, Ching-Te Chiu*

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

ARTICLE INFO

Keywords:
Deep convolutional model compression
Action recognition
Knowledge distillation
Transfer learning

ABSTRACT

Recently, Convolutional Networks have great progress in classifying images. While action recognition is different from still image classification, video data contains temporal information which plays an important role in video understanding. Currently most CNN-based approaches for action recognition has excessive computational costs, an explosion of parameters and computation time. The most efficient method currently trained a deep network directly on the compressed video contains the motion information. However, this method has a large number of parameters. We propose a multi-teacher knowledge distillation framework for compressed video action recognition to compress this model. With this framework, the model is compressed by transferring the knowledge from multiple teachers to a single small student model. With multi-teacher knowledge distillation, students learn better than single-teacher knowledge distillation. Experiments show that we can reach a 2.4 × compression rate in number of parameters and 1.2 × computation reduction with 1.79% loss of accuracy on the UCF-101 dataset and 0.35% loss of accuracy on the HMDB51 dataset.

1. Introduction

Traditional classification methods such as support vector machine (SVM) are commonly used [1–3], however, due to the rapid development of deep learning and the huge amount of datasets, neural network based learning approaches become more and more popular. Human action recognition has been an active research topic in computer vision due to its wide range of applications, such as smart home and driver monitoring. Implementation of these applications using VLSI or embedded computing systems has low-power and real-time requirements. Recently, Convolutional Networks have great progress in classifying images, ConvNets have also been considered to solve action recognition problem. Most current CNN-based approaches for action recognition are based on two-stream [4] and C3D [5].

For two-stream-based approaches, the input to the spatial and temporal streams are RGB frames and stacks of multiple-frame dense optical flow fields, respectively. Using dense optical flow information for action recognition usually has good accuracy, but it has excessive computational costs.

C3D-based approaches learn spatio-temporal feature with clips of multiple continuous frames, their architecture contains 3D convolution and fully-connected layer cause an explosion of parameters and computation time.

These above methods are unable to perform action recognition efficiently. Some approaches explored other robust deep video representations [6,7], CoViAR [7] train a deep network directly on the compressed video. The video compression techniques (like MPEG, H.264 etc.) retain only a few frames completely and reconstruct other frames based on offsets from the complete images, called motion vectors and residual error. They avoid to calculate the dense optical flow due to the motion vector and still achieve good performance. They also achieve best efficiency, while requiring a far lesser amount of data. However, CoViAR has excessive storage size due to the number of parameters. For embedded mobile applications, their size consumes excessive storage/memory and computational resources. Therefore, the model size reduction becomes crucial.

In our work, we propose a multi-teacher knowledge distillation framework to compress CoViAR model. We improve the accuracy after compression with the comprehensive knowledge from multiple teachers. We also propose two restrictions in data preprocessing to increase the accuracy. The operation flow in our proposed method is shown in Fig. 1

2. Related works

In this section we provide a brief overview about action recognition and model compression.

2.1. Action recognition

The approaches of action recognition can be divided into two categories. One is traditional hand-crafted features, such as Histogram of Oriented Gradients (HOG) [8] and Histogram of Optical Flow (HOF)

* Corresponding author.
E-mail address: chiusms@cs.nthu.edu.tw (C.-T. Chiu).

https://doi.org/10.1016/j.sysarc.2019.101695
Received 20 September 2018; Received in revised form 27 November 2019; Accepted 28 November 2019
Available online 4 December 2019

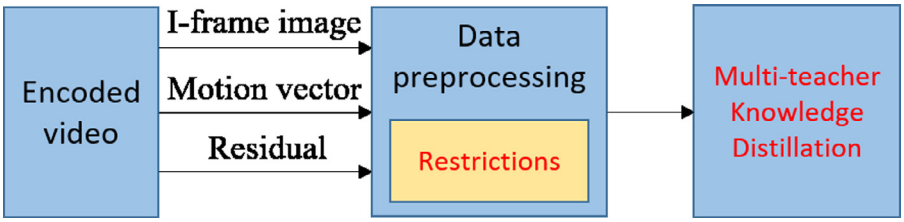

Fig. 1. The operation flow in our proposed method.

[9]. These methods consider independent interest points across frames, later aggregation based on dense trajectories have been used [10–12]. Among them, iDT [12] is competitive even today.

The other category is deep learning. In the past few years, convolutional neural networks (CNN) achieve impressive performance for natural image classifications. While action recognition is different from still image classification, video data contains temporal information which plays an important role in video understanding. Recently several attempts have been made to go beyond individual image-level appearance information and exploit the temporal information using ConvNet architectures. Most current approaches for action recognition are based on two-stream [4] and C3D [5].

Two-stream based approach [4,13–16] contain two separate recognition streams (spatial and temporal), which are then combined by fusion. The spatial stream performs action recognition from still video frames, while the temporal stream recognizes action from motion in the form of dense optical flow. The dense optical flow has excessive computational costs although it is a well representation of motion information. There are some other approaches to exploring how to fuse spatial and temporal features [17–22].

3D convolution based approach [5,23,24] explored 3D ConvNets on video datasets, where they tried to learn both spatial and temporal features with 3D convolution operations. C3D [5] contain 3D convolution and fully-connected layers causes an explosion of parameters and computation time. Res3D [23] is proposed after C3D work, it outperforms C3D, while being two times faster at inference time, two times smaller in model size. But Res3D still has high computation due to 3D convolution operations. I3D [24] is a Two-Stream Inflated 3D ConvNet (I3D), I3D models being much deeper, while having much fewer parameters than the C3D-like model. But I3D is trained on 64-frame video snippets (64 RGB frames, 64 optical flow frames) while causing an explosion of input data size and using dense optical flow causes excessive computational costs.

There are some approaches exploring other robust deep video representations, MV-CNN [6] proposed to accelerate the Two-Stream architecture [4] by replacing dense optical flow with motion vector which can be obtained directly from compressed videos without extra calculation. However, motion vector extracted from compressed videos contains lots of noisy movement information and it is much coarser than optical flow. In spite of accelerating the Two-Stream architecture, it lost accuracy.

Motivated by that fact that the data amount of compressed video is much less than the raw video, Wu et al. [7] (CoViAR) proposed to train a deep network directly on the compressed video. Their model consists of three CNNs, each of which learns to model one kind of representation in a compressed video: I-frame images, P-frame motion vectors and P-frame residuals. Since the I-frame containts most of the information. They use Res-Net-152 to model I-frames and Res-Net-18 to model P-frame motion vectors and residuals. Their architecture is shown in Fig. 2, all networks can be trained independently and their scores are fused for final prediction. CoViAR used Temporal Segments [13] to capture long term dependency, it extracts short snippets over a long video sequence with a sparse sampling scheme to enable efficient and effective learning using the whole action video. They avoid to calculate the optical flow explicitly and still achieve good performance. Based on the fact that video compression reduces irrelevant information from the data and the increased relevance and reduced dimensionality makes computation effectively [7] (CoViAR). They also achieve high efficiency, while requiring a far lesser amount of data. However, CoViAR uses one Res-Net-152 for I-frame images so it has excessive storage size due to the big amount of parameters in Res-Net-152. The compressed characteristic of I-frame is not utilized while selecting the CNN model.

2.2. Model compression

The number of parameters get larger as deep neural networks get deeper, deploying a network on a small device becomes increasingly difficult. Thus reduction in amount of parameters and faster computation are highly desired.

In general, methods of compressing the deep convolutional models can be separated into four main categories: network pruning [25–28], matrix decomposition [29–32], parameter sharing [33,34] and knowledge distillation [35–41].

Network pruning is to prune on the weights which are not important or are even redundant. Pruning unimportant weights layer by layer, and fine-tune after pruning every time. Methods of this category are likely to spend lots of time on iteratively pruning and fine-tuning.

Matrix decomposition is to do the low-rank decompositions on the weight matrices, such as using the singular value decomposition or tensor decomposition. Methods of this category can reach an impressive compression rate on the fully-connected layers.

Parameter sharing is to use hashing functions or clustering to group the weights in each layer. Quantized CNN [34] first learned several sub-codebooks for each layer and then used the codebooks to quantized the weights.

Knowledge distillation is a different kind of training to transfer the knowledge from the cumbersome model to a small model. The small model has richer knowledge than vanilla small model yet own less parameters and complexity than original cumbersome models. We utilize Knowledge Distillation [35] in our proposed method to compress model size.

3. Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks

Distillation (Hinton et al. [35]) is the technique that transfer the knowledge from the cumbersome model to a small model, we call them teacher and student model, respectively. The student model has richer knowledge than vanilla student model yet own less parameters and complexity than original teacher models. An overview of knowledge distillation is shown in Fig. 3. Distillation use the class probabilities produced by the teacher model as ”soft labels” for training the student model.

3.1. Knowledge distillation

Compressed video can reduce the irrelevant information from the raw video data. Comparing with the raw RGB video, the compressed I-frames contain the fundamental information in reduced dimensions so I-frames should be able to be modelled by a small CNN model rather than Res-Net-152. In order to improve the above problem, we apply the knowledge distillation to CoViAR by integrating the knowledge from

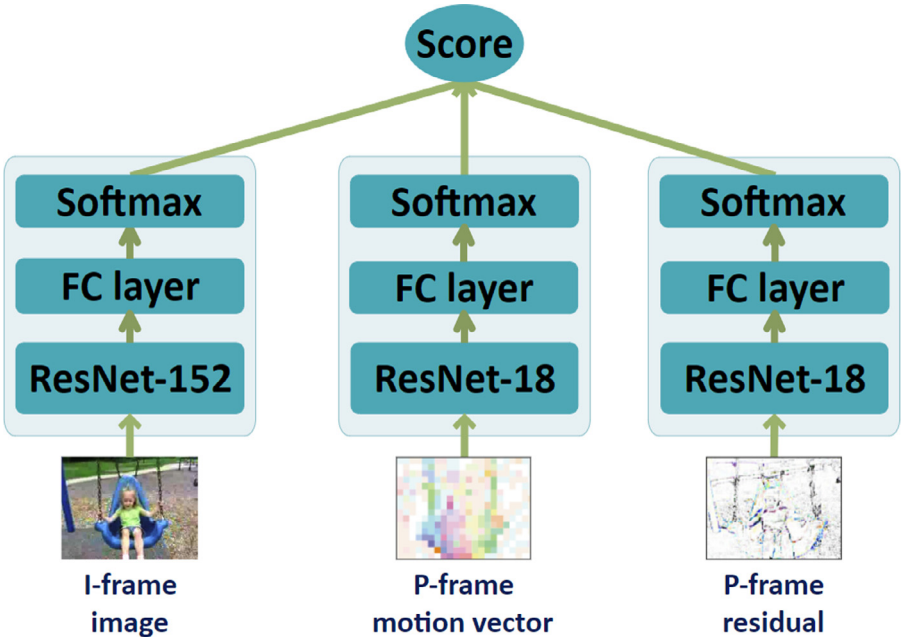

Fig. 2. CoViAR architecture. All networks can be trained independently. Models are shared across P-frames. [7].

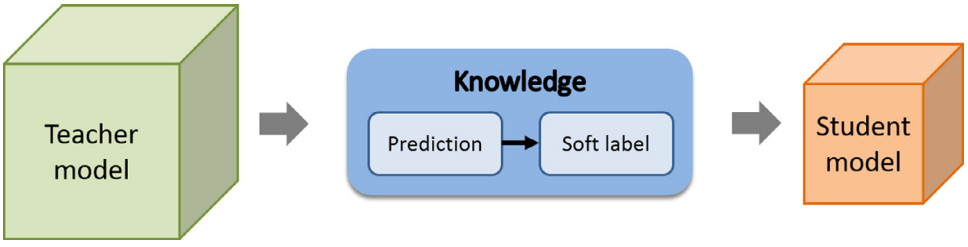

Fig. 3. An overview of knowledge distillation.

multiple CNNs, then train the student models with distillated knowledge. We not only reduce the biggest model size of CNN which model image representation, but also make other CNNs which model motion vectors and residuals learn better.

Knowledge is transferred from the teacher model to the student by making the distribution of class probabilities of the student as close as that predicted by the teacher model. This is done by minimizing a loss function on the output of a softmax function at the teacher model's logits. However, in many cases, this probability distribution has the correct class at a very high probability, with all other class probabilities very close to zero. As such, it doesn't provide much information beyond the ground truth labels already provided in the dataset. Fig. 4 shows the concept of knowledge distillation. As shown in Fig. 4, the total loss consists of two parts. One loss is generated from comparing the soft labels between the teacher model and student model. The other loss is produces from comparing the hard labels between the student model and ground truth layers. We describe the soft labels and hard labels blow. To tackle this issue, Hinton et al., 2015 [35] introduced the concept of "softmax temperature". The probability distribution generated by the softmax temperature function becomes softer, providing more information as to which classes the teacher found more similar to the predicted class and we call them soft labels. The soft labels are embedded in the teacher model, and these knowledge in soft labels are transferring to the student model. In addition to the soft labels, Hinton found that it is beneficial to have the loss of comparing the student's predicted class probabilities and the ground-truth labels and we call this hard labels.

For video-level tasks, CoViAR [7] uses sparse sampling strategy [13] on an input video, and average the features across $k = 3$ sampled segments during training. In our proposed multi-teacher knowledge distillation framework, the logits vector produced by student network for an input video $v_i, i = 1, \ldots, N$ is represented by $(z_s)_i$, where the dimension of vector $(z_s)_i = [(z_s)_i^1, \ldots, (z_s)_i^C]$ is the number of categories C. The softmax layer converts logits vector $(z_s)_i$ to probability distribution $(q_s)_i = [(q_s)_i^1, \ldots, (q_s)_i^C]$,

$$(q_s)_i = Softmax((z_s)_i), \tag{1}$$

where

$$(q_s)_i^j = \frac{\exp\left((z_s)_i^j\right)}{\sum_k \exp\left((z_s)_i^k\right)} \quad \text{for } j = 1, \ldots, C \tag{2}$$

On the other hand, the logits vector produced by teacher network for an input video $v_i, i = 1, \ldots, N$ is represented by $(z_t)_i$, where the dimension of vector $(z_t)_i = ((z_t)_i^1, \ldots, (z_t)_i^C)$ is the number of categories C. By introducing a parameter called temperature T, the generalized softmax layer *GSoftmax* converts logits vector $(z_t)_i$ to soft probability distribution $(q_t^T)_i = [(q_t^T)_i^1, \ldots, (q_t^T)_i^C]$,

$$\left(q_t^T\right)_i = GSoftmax((z_t)_i, T), \tag{3}$$

where

$$\left(q_t^T\right)_i^j = \frac{\exp\left((z_t)_i^j/T\right)}{\sum_k \exp\left((z_t)_i^k/T\right)} \quad \text{for } j = 1, \ldots, C \tag{4}$$

Distillation use the class probabilities produced by the teacher model as "soft labels" for training the student model.

There are two objective functions when training the student model. The first objective function $\mathcal{L}_1$ is to minimize the cross entropy with the soft labels $(q_t^T)_i$ and the soft probability $(q_s^T)_i$ produced by the student model. $(q_s^T)_i$ is computed by *GSoftmax* with the same temperature T as teacher model,

$$\left(q_s^T\right)_i = GSoftmax((z_s)_i, T), \tag{5}$$

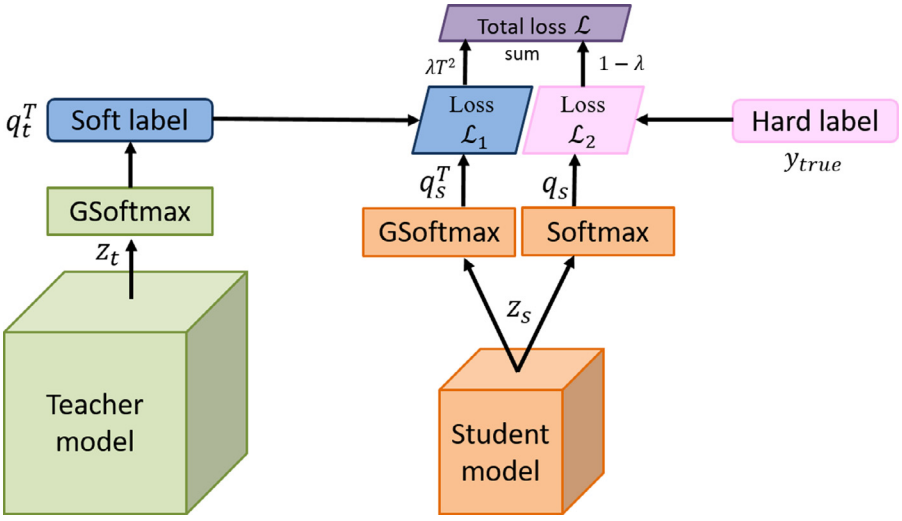

Fig. 4. The schematic of the knowledge distillation setup.

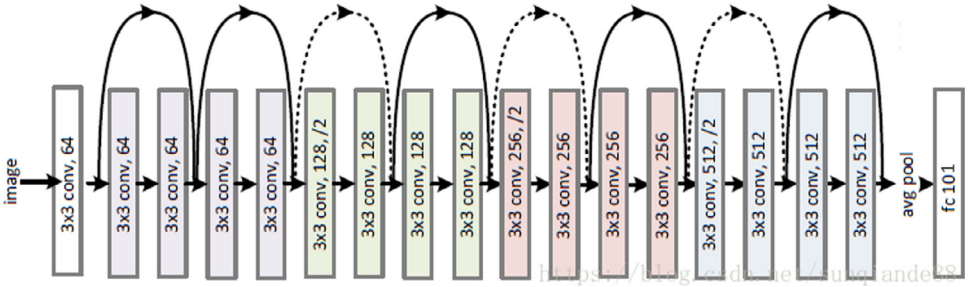

Fig. 5. The architecture of ResNet-18.

where

$$\left(q_s^T\right)_i^j = \frac{\exp\left((z_s)_i^j/T\right)}{\sum_k \exp\left((z_s)_i^k/T\right)} \quad \text{for } j = 1, \ldots, C \tag{6}$$

The first objective function $\mathcal{L}_1$:

$$\arg\min_W \mathcal{L}_1(W) = \arg\min_W -\frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C}(q_t^T)_i^c \ln\left(q_s^T\right)_i^c \tag{7}$$

where $(q_s^T)_i^c$ produced by student is the probability that the ith video belongs to cth class, $(q_t^T)_i^c$ is the soft label produced by teacher, W is the weights of student, N is the amount of the training videos, C is the amount of the total classes.

The second objective function $\mathcal{L}_2$ is to minimize the cross entropy with the hard labels y_{true} and the probability $(q_s)_i$ produced by student.

$$\arg\min_W \mathcal{L}_2(W) = \arg\min_W -\frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C}(y_{true})_i^c \ln(q_s)_i^c \tag{8}$$

where $(q_s)_i^c$ produced by student is the probability that the ith video belongs to cth class, $(y_{true})_i^c$ is the hard label information, $(y_{true})_i^c = 1$ if ith video belongs to cth class, otherwise $(y_{true})_i^c = 0$. W is the weights of student, N is the amount of the training videos, C is the amount of the total classes.

The overall objective function $\mathcal{L}$ is a weighted average of two different objective functions. The schematic of the knowledge distillation setup is shown in Fig. 4.

$$\arg\min_W \mathcal{L}(W) = \arg\min_W \lambda T^2 \mathcal{L}_1(W) + (1-\lambda)\mathcal{L}_2(W) \tag{9}$$

where W is the weights of student and λ is a relative weight.

Table 1
The inference time of CoViAR separated models.

CoViAR	ResNet-152 on I-frame	ResNet-18 on motion vector	ResNet-18 on residual
Inference time (ms)	1.35	0.33	0.24

3.2. Distilling on given input I-frame

In CoViAR architecture, the spatial network ResNet-152 spent more time than other temporal networks. The inference time of CoViAR separated models is shown in Table 1. Because above reason, we decided to begin with distilling the knowledge of spatial network to a smaller model. Fig. 6 shows the single-teacher distillation on I-frame image. According to ResNet architecture for ImageNet [42], the number of parameters of ResNet-152 is about 58.2 million and ResNet-18 is about 11.2 million, the computational cost of ResNet-152 is 11.3 GFLOPs and ResNet-18 is 1.8 GFLOPs. The spatial network will have 5.2 times compression rate and 6.28 times computation reduction due to model compression from 152 layers to 18 layers.

After single-teacher distillation, we found that the distilled network lost accuracy after compression. Loss of accuracy was no doubt a necessary consequence of reducing number of parameters. Due to this observation we propose to teach the student more comprehensive knowledge which from multiple teachers with different input types in an attempt to increase accuracy.

In our proposed Multi-teacher Distillation, the candidates of teachers are from CoViAR separated models. When we select all three input types: I-frame image, motion vector and residual as teachers, an overview of three-teacher distillation is shown in Fig. 7. We integrate the knowledge from multiple teachers and teach the student this comprehensive

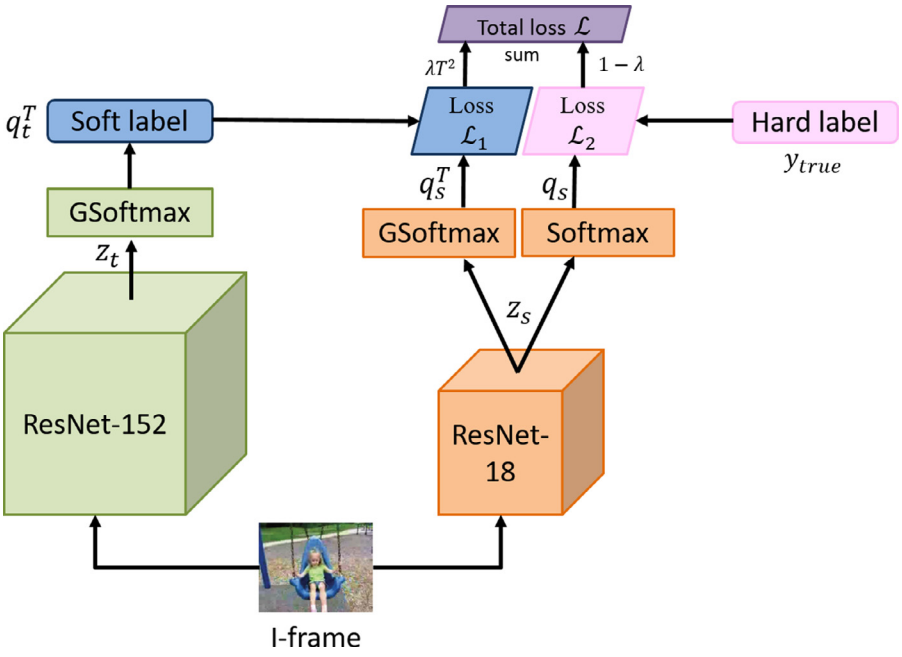

Fig. 6. The architecture of single-teacher distillation on I-frame image.

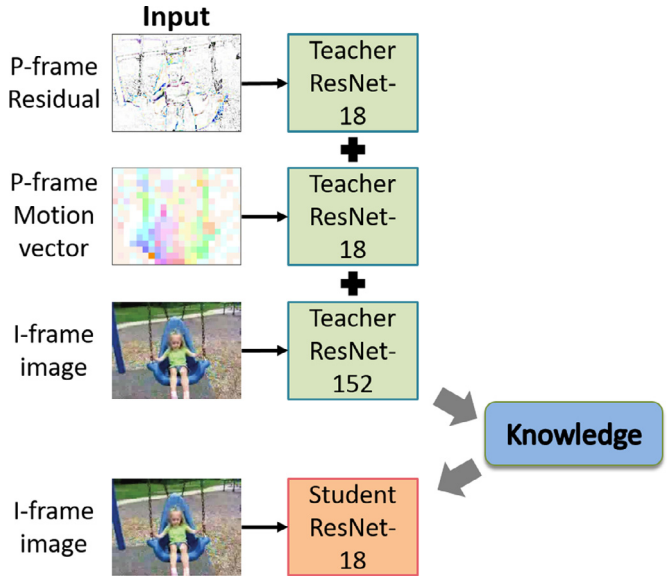

Fig. 7. An overview of three-teacher distillation on I-frame image.

knowledge in the form of the soft label. The soft label is a weighted average of different soft probability distributions from multiple teachers. For three teachers case, the teacher $t1$, $t2$ and $t3$ produce soft probability distributions q_{t1}^T, q_{t2}^T and q_{t3}^T with *GSoftmax* layer and the same temperature T. The soft label q_t^T will be a weighted average of q_{t1}^T, q_{t2}^T and q_{t3}^T,

$$q_t^T = \frac{\mu_1 \times q_{t1}^T + \mu_2 \times q_{t2}^T + \mu_3 \times q_{t3}^T}{\mu_1 + \mu_2 + \mu_3} \tag{10}$$

where q_{t1}^T, q_{t2}^T and q_{t3}^T are weighted by μ_1, μ_2 and μ_3, respectively. The architecture of three-teacher distillation on I-frame image is shown in Fig. 8.

3.3. Distilling on given input P-frame

Although the temporal networks which take motion vector and residual as input have the smallest model size in ResNet series, they can also be transferred the more comprehensive knowledge by distillation. For the student with given input motion vector, Fig. 9 show that we select all three input types: I-frame image, motion vector and residual as teachers. Similarly, for the student with given input residual, Fig. 10 show the overview of three-teacher distillation on input residual.

3.4. Multi-teacher to multi-student mode

We utilize the knowledge distillation technique not only to compress the spatial network model but also promote the performance of temporal networks by multi-teacher knowledge distillation. Fig. 11 shows the comprehensive knowledge from multiple teachers was taught to different students with different input type separately, and then fuse the results of separated students for final prediction.

4. Experimental results

We implement our proposed architecture by using the open source PyTorch [43]. Our models are pre-trained on the ILSVRC2012-CLS dataset [42], random, and we optimize our architecture by using mini-batch and Adam optimizer [44] with weight decay of 0.0001, eps of 0.001, initial learning rate of 0.003 for input I-frame image, learning rate of 0.01 for input motion vector, learning rate of 0.005 for input residual, which is divided by 10 when the accuracy plateaus. We train or evaluate on a server with Intel i7-7800K 3.50GHz CPU, 16 GB memory and NVIDIA GeForce GTX 1080 GPU.

4.1. Training skills

4.1.1. Data preprocessing

The data preprocessing technique we used is following CoViAR [7]. We use MPEG-4 coding format video, which have on average 11 P-frames for every I-frame. Following TSN [13], Videos are resized to 340 × 256. The input data: images, motion vectors and residuals are extracted from the resized encoded videos.

There are two restrictions on input source while distilling. First, for single teacher case, the same image is inputted into both teacher and student model. For multi-teacher case, the extracted data must be from the same frame or from the same GOP (Group of pictures). Because the accumulated motion vector and residual from each P-frame depend on

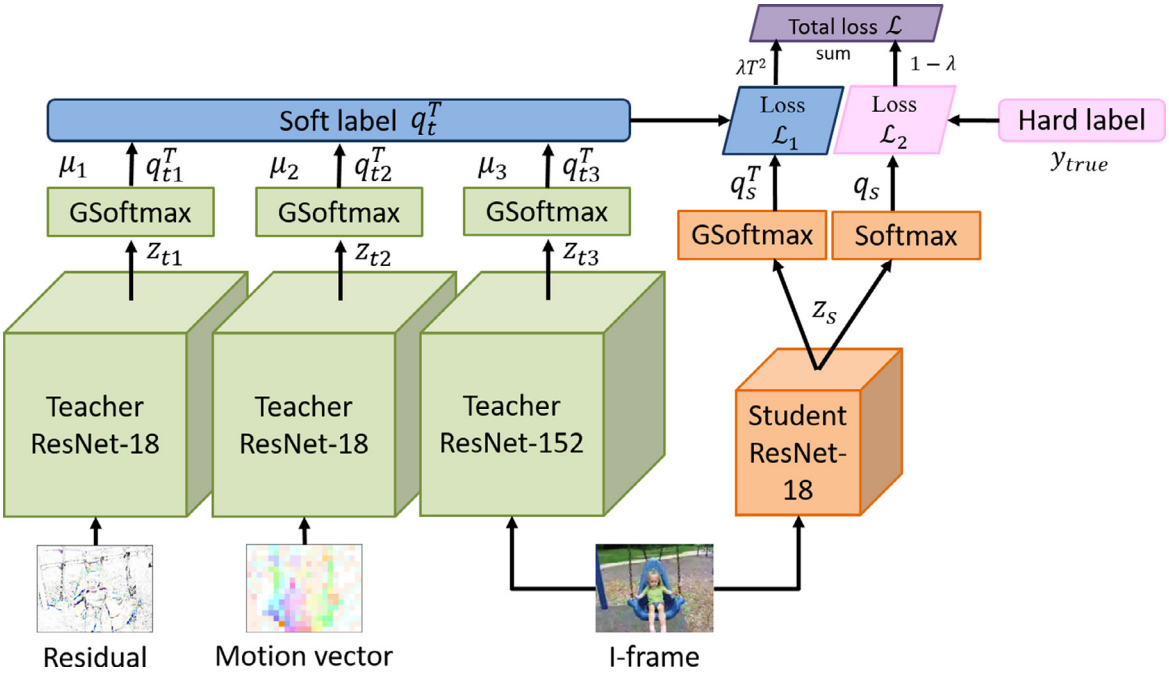

Fig. 8. The architecture Multi-teacher distillation on I-frame image.

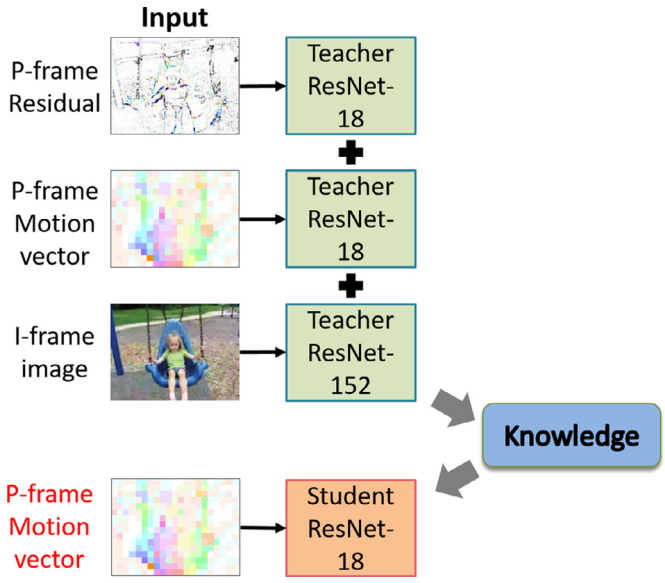

Fig. 9. An overview of three-teacher distillation on P-frame motion vector.

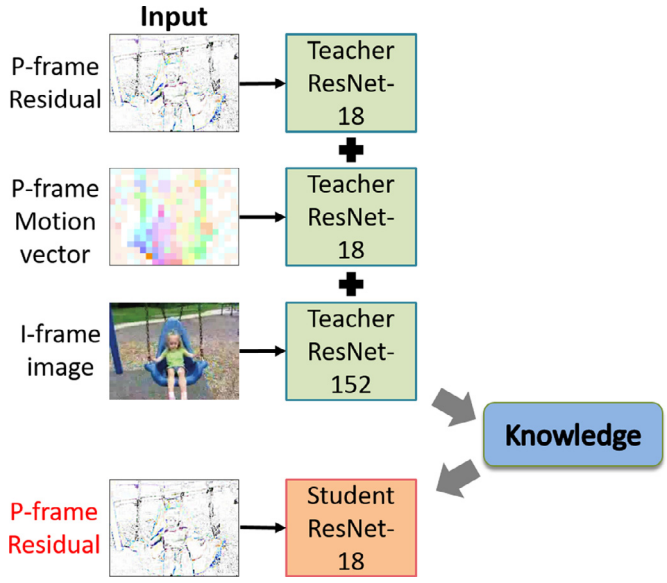

Fig. 10. An overview of three-teacher distillation on P-frame residual.

the last I-frame. The teachers observe the data from the same GOP will make the student learn better. Second, the input data for teachers and student must have the same preprocessing process (data augmentation), because different preprocessing process may affect teachers' observation. These restrictions are shown in Fig. 12. Our experiment show that the performance of distilled student model was promoted with these restrictions.

4.1.2. Data augmentation

It is necessary to provide enough training data to achieve satisfactory training results for CNN models. To increase the amount of training data, in the training stage, we adopt the data augmentation techniques following Wang *et al.* [13] before the data input into the CNNs. The data augmentation process includes three steps: color jittering, corner cropping plus scale-jittering and random horizontal flipping.

4.2. Dataset and evaluation protocol

We evaluate our method on two action recognition datasets, UCF-101 [45] and HMDB-51 [46]. UCF-101 contains 13,320 videos from 101 action categories. HMDB-51 contains 6766 videos from 51 action categories. Each video in both dataset is annotated with one action label. Each dataset has 3 training/testing splits for evaluation. We report the average performance of the 3 testing splits.

During testing we uniformly sample 25 frames, each with flips plus 5 corner crops, and then average the scores for final prediction following CoViAR [7].

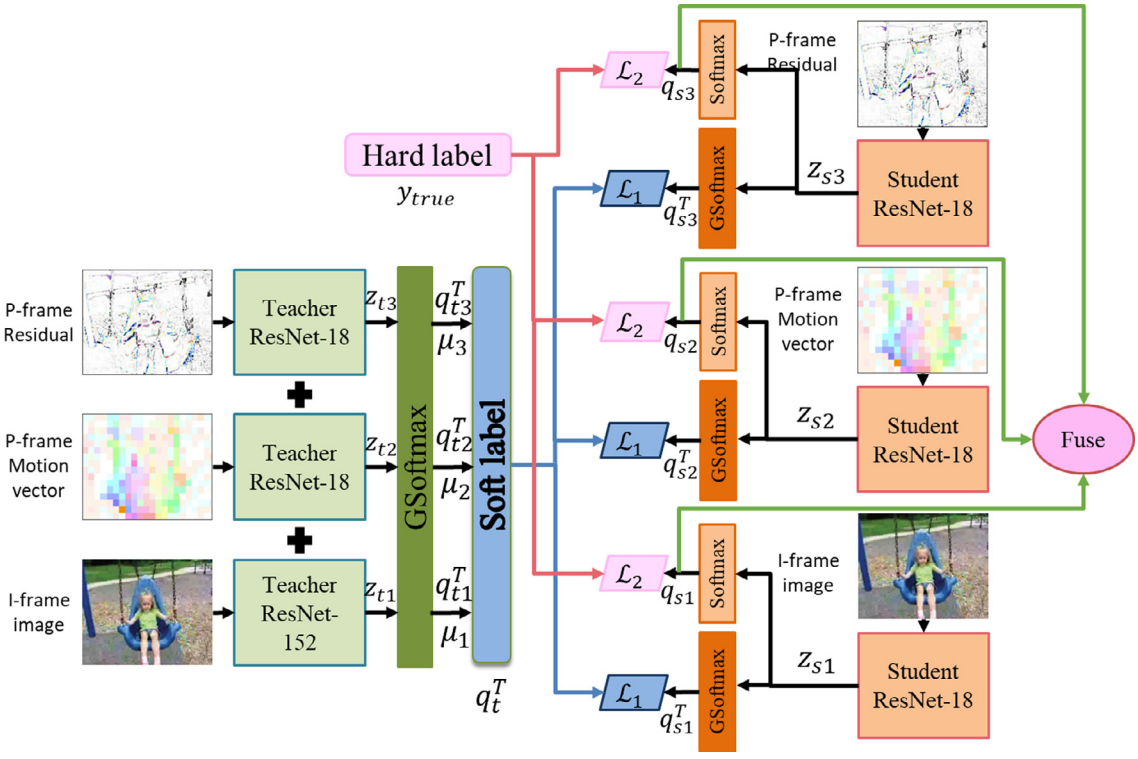

Fig. 11. The architecture of Multi-teacher to Multi-student mode.

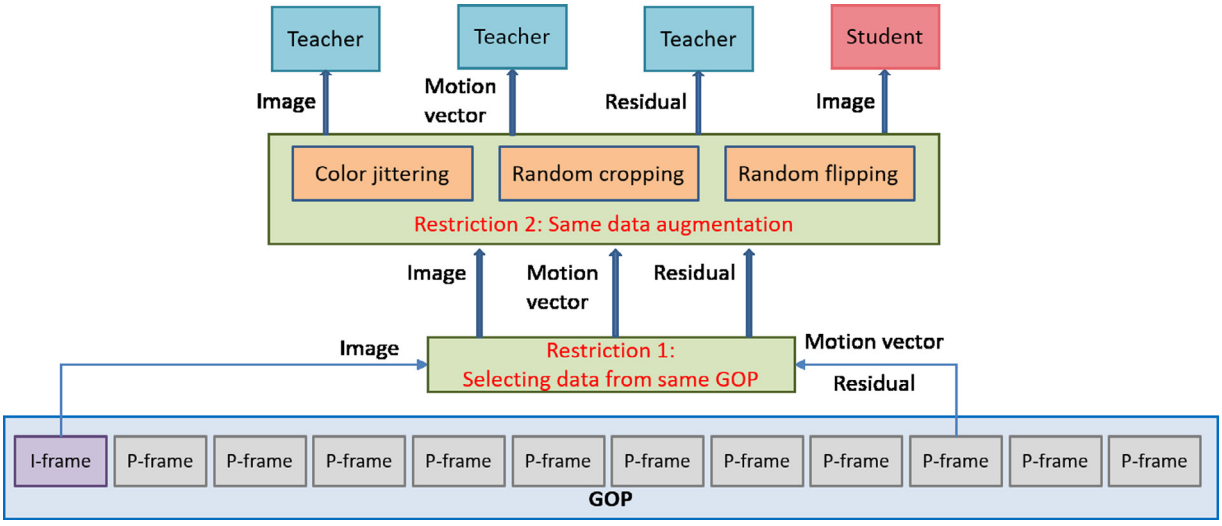

Fig. 12. The restrictions in data preprocessing.

4.3. Training and testing results

4.3.1. Evaluation results for distilling on given input I-frame

Before transferring the knowledge of teacher model ResNet-152 to the student model ResNet-18 which has the same input as teacher. We evaluate the distilling hyperparameters temperature T and relative weight λ firstly and show in Figs. 13 and 14. In the condition of 50 training epochs, using $T = 10$ and $\lambda = 0.3$ achieves highest accuracy on UCF-101 dataset, using $T = 5$ and $\lambda = 0.7$ achieves highest accuracy on HMDB51 dataset. In the following experiments, the same hyperparameter setting is used. With further observation, low temperature gets better accuracies with higher weight λ, and high temperature gets better accuracies with lower weight λ.

Next, we transfer the knowledge of teacher model ResNet-152 to the student model ResNet-18 which has the same input as teacher. The Single-teacher Distillation on I-frame evaluation results are shown in Table 2. On UCF-101, ResNet-152 achieves 87.31% with self-learning. ResNet-18 achieves 78.64% with self-learning. Our Single-teacher Distillation on I-frame achieves 84.38% by distilling the knowledge of ResNet-152, which has 2.9% loss comparing to ResNet-152 and 5.74% gain comparing to self-learning ResNet-18. This experiment indicates that the knowledge of ResNet-152 was successfully transferred to ResNet-18 model. Note that for better comparison, we set the same learning rate 0.0003 here which following CoViAR [7].

We found that the distilled network lost accuracy after compression. Due to this observation we propose to teach the student more

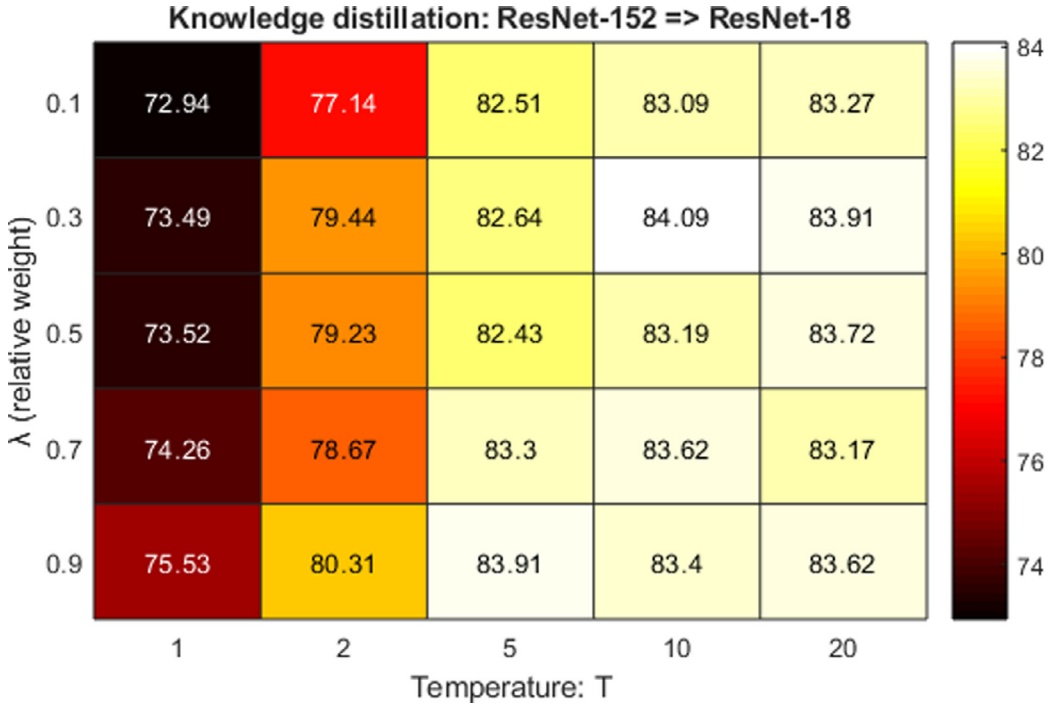

Fig. 13. Evaluation of distilling hyperparameters temperature T and relative weight λ on UCF-101 dataset. Training epochs = 50.

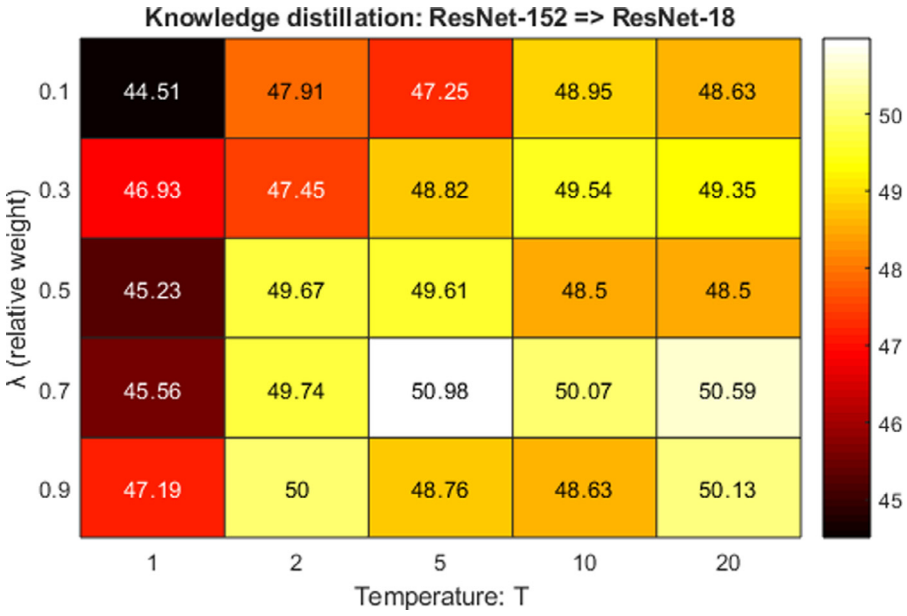

Fig. 14. Evaluation of distilling hyperparameters temperature T and relative weight λ on HMDB51 dataset. Training epochs = 50.

Table 2
The accuracy of Single-teacher distillation on I-frame. For better comparison, we set the same learning rate 0.0003 which following CoViAR [7].

Distilling	Input data	Architecture	Accuracy(%)(split 1)	
			UCF-101	HMDB51
Self-learning	I-frame	ResNet-152	87.31	52.29
Self-learning		ResNet-18	78.64 (-8.67)	46.67 (-5.62)
Distilling		ResNet-18	84.38 (-2.9)	48.43 (-3.86)

comprehensive knowledge which from multiple teachers with different input type. Table 3 shows the experiments of multi-teacher distillation on I-frame. The experiments demonstrate that single-teacher knowledge distillation outperform multi-teacher on input I-frame. We argue that most of the information is stored in I-frames, integrating other different knowledge will cause confusion.

The restrictions on input data source also have profound impact on multi-teacher knowledge distillation. First restriction: the extracted data must be from the same frame or from the same GOP (Group of pictures). Second restriction: the input data for teachers and student has the same preprocessing process (data augmentation). We experiment on three-teacher distillation on given input I-frame and show in Table 4. Without the restrictions, the student only gets 81.55% in accuracy. With restriction 1, the student gains 1.53% improvement in accuracy. With both restrictions, the student gains 3.51% improvement in accuracy. The experiments indicate the restrictions we propose have great influence on multi-teacher knowledge distillation.

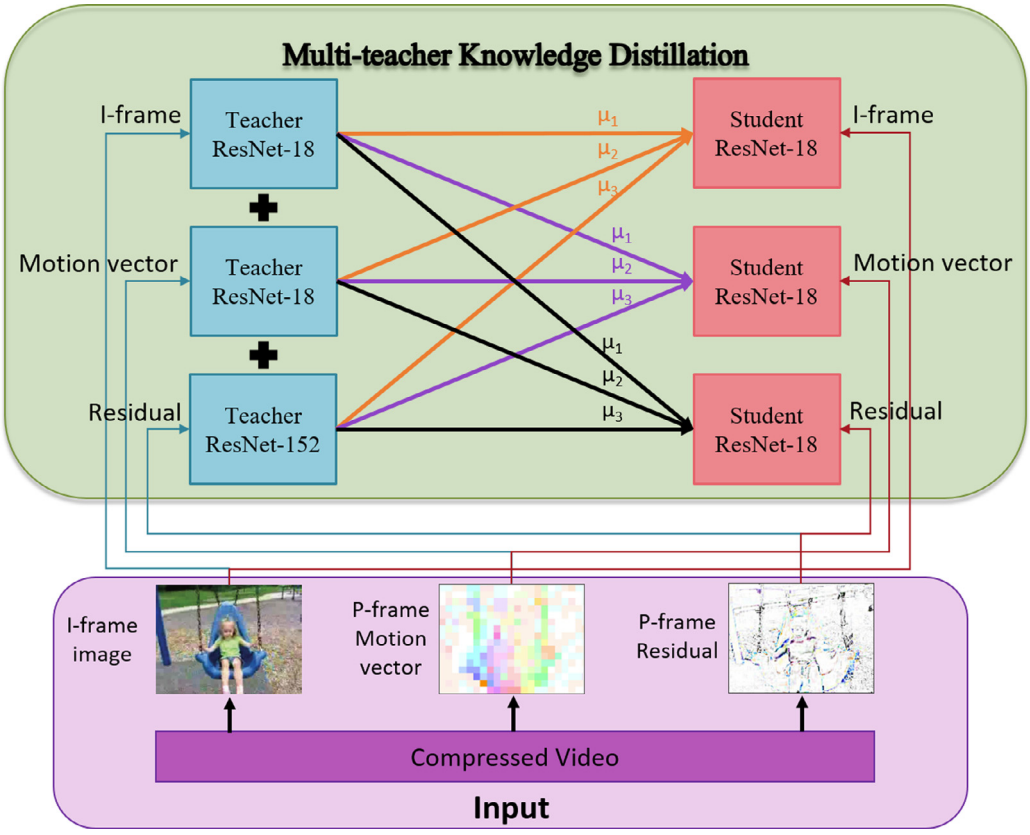

Fig. 15. An overview of multi-teacher distillation on different input type.

Table 3
The accuracy(%) of Multi-teacher distillation on student I-frame.

Teacher	Student	UCF-101			HMDB51		
		Split 1	Split 2	Split 3	Split 1	Split 2	Split 3
I-frame	I-frame	**85.22**	**83.61**	83.20	**50.72**	**47.71**	**48.30**
I-frame + mv		84.14	83.18	83.23	48.24	46.99	48.04
I-frame + residual		84.83	82.78	82.79	50.20	45.36	46.80
I-frame + mv + residual		85.06	82.46	**83.69**	50.39	45.95	46.99

Table 4
Compare accuracies between different input preprocessing process in the case distilling teachers (I-frame + mv + residual) to the student (I-frame).

Restriction	Accuracy
No restriction	81.55%
Restriction 1	83.08% (+1.53)
Restriction 1 + 2	**85.06%** (+3.51)

4.3.2. Evaluation results for distilling on given input P-frame

Tables 5 and 6 shows the experiments of multi-teacher knowledge distillation on given input P-frame. The experiments demonstrate that the students gain better performance with more comprehensive knowledge.

4.3.3. Evaluation results for multi-teacher to multi-student mode

In Fig. 15, we utilize the knowledge distillation technique not only to compress the spatial network model but also promote the performance of temporal networks by multi-teacher knowledge distillation. We compare Single-teacher and multi-teacher knowledge distillation in Table 7. In single-teacher case, the score of distilled spatial network was fused with original CoViAR temporal networks. In multi-teacher case, we select the best performance of distilled network from different input types. The experiment indicate that multi-teacher distillation will improve the

Table 5
The accuracy(%) of Multi-teacher distillation on P-frame motion vector.

Teacher	Student	UCF-101			HMDB51		
		Split 1	Split 2	Split 3	Split 1	Split 2	Split 3
I-frame	mv	65.50	68.88	66.69	31.24	35.49	36.67
I-frame + mv		**69.34**	69.44	70.91	43.92	**43.53**	**46.27**
I-frame + mv + residual		67.51	**71.13**	**72.05**	**44.77**	41.76	45.56

Table 6
The accuracy of Multi-teacher distillation on P-frame residual.

Teacher	Student	UCF-101			HMDB51		
		Split 1	Split 2	Split 3	Split 1	Split 2	Split 3
Self-learning	residual	80.04	80.80	81.79	42.68	37.84	42.03
I-frame + residual		**83.45**	82.59	**84.47**	49.80	41.31	48.10
I-frame + mv + residual		82.92	**83.85**	84.09	**50.98**	**45.03**	**49.15**

Table 7
Comparison between single and multi-teacher distillation.

	Input type	UCF-101	HMDB51
Single-teacher distillation	I-frame	84.40%	48.91
	mv	65.50%	34.47
	Residual	80.04%	40.85
	Fusion	88.29%	55.27
Multi-teacher distillation	I-frame	85.06% (+0.66)	48.91
	mv	69.34% (+3.84)	44.86 (+10.39)
	Residual	83.45% (+3.41)	48.39 (+7.54)
	Fusion	88.87% (+0.58)	56.16 (+0.89)

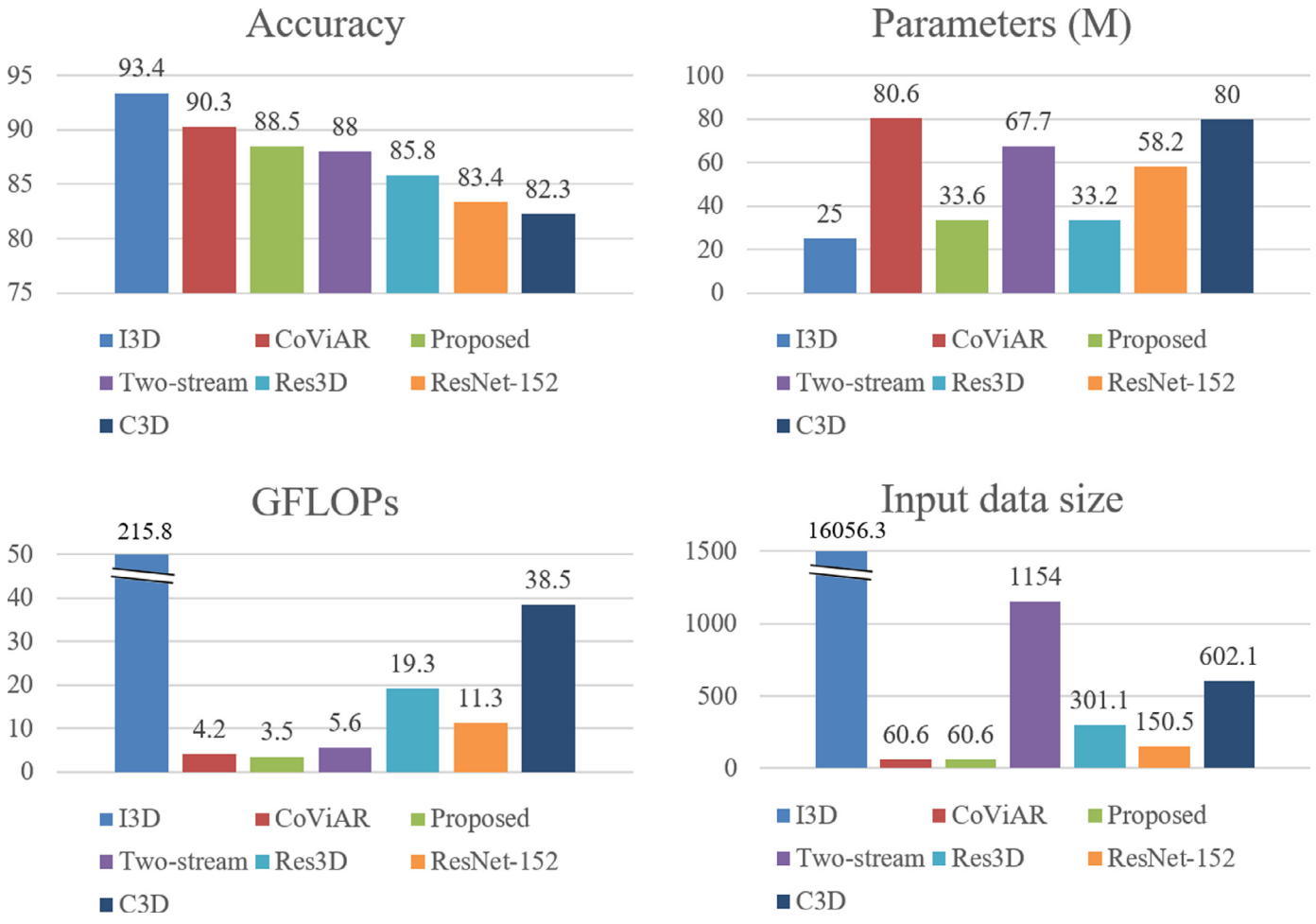

Fig. 16. Network computation complexity and accuracy on UCF-101.

accuracy after compression, note that I-frame has the biggest impact on fusion result.

We compare multi-teacher distillation with uncompressed CoViAR [7] model in Table 8. By multi-teacher knowledge distillation, we compress the spatial network but having 3.08% loss in accuracy, the temporal network on input motion vector has 3.82% increase of accuracy and another temporal network on input residual has 3.04% increase of accuracy. The final result after compression is shown in Table 9, we achieve 2.4x compression rate on the number of parameters with 1.79% loss of accuracy on the UCF-101 dataset and 0.35% loss of accuracy on the HMDB51 dataset.

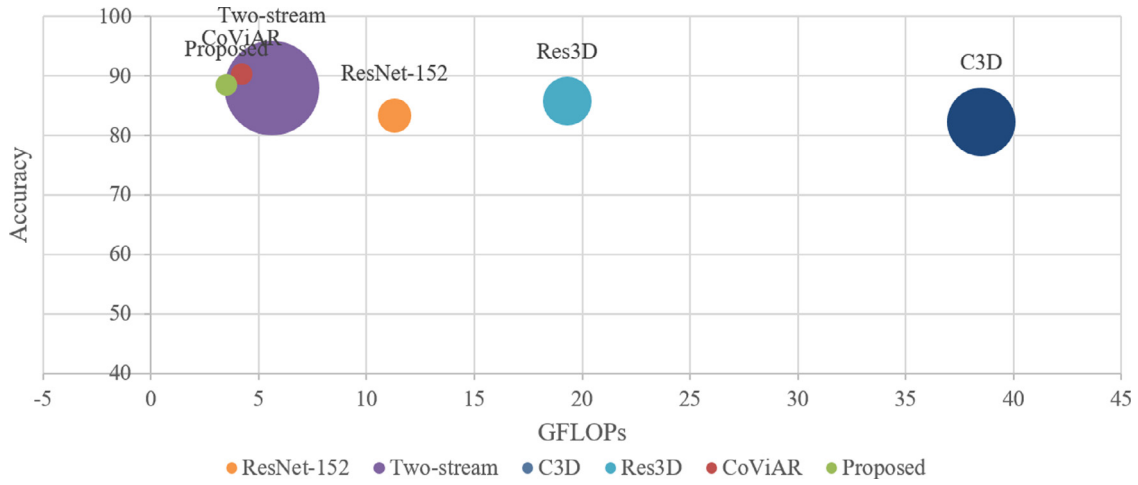

Fig. 17. Network computation complexity and accuracy on UCF-101. Node size denotes the input data size.

Table 8
The accuracy of Multi-teacher to Multi-student mode. The top half of the table is the baseline CoViAR, the bottom half of the table is the accuracy of Fusing the distillations on different inputs.

Baseline CoViAR	Input	Architecture	UCF-101	HMDB51
	Iframe	ResNet-152	87.25	51.13
	mv	ResNet-18	67.02	34.47
	residual	ResNet-18	80.88	40.85
	Fusion		90.29	56.51
Multi-teacher Distillation	Student	Architecture	UCF-101	HMDB51
	Iframe	ResNet-18	84.17 (−3.08)	48.91
	mv	ResNet-18	70.84 (+3.82)	44.86 (+10.39)
	residual	ResNet-18	83.92 (+3.04)	48.39 (+7.54)
	Fusion		88.50 (−1.79)	56.16 (−0.35)

Table 9
The final result after compression.

	UCF-101	HMDB51	Parameters (M)	GFLOPs	inference time (ms)
CoViAR	90.29	56.51	80.64	4.2	12.88
Proposed	88.50 (−1.79)	56.16 (−0.35)	33.6 (0.42x)	3.5 (0.83x)	6.88 (0.53x)

Table 10
Network computation complexity and accuracy on UCF-101 dataset. Asterisk indicates results evaluated only on the UCF-101 dataset, these methods can achieve higher accuracy with other large-scale video datasets.

	Accuracy	Parameters (M)	GFLOPs	Input data size
ResNet152 [47]	83.4%	58.2 (0.72 ×)	11.3 (2.69 ×)	150.5 K (2.48 ×)
Two-stream [4]	88.0%	67.7 (0.84 ×)	5.6 (1.33 ×)	1154.0 K (19.04 ×)
C3D [5]*	82.3%	80.0 (0.99 ×)	38.5 (9.17 ×)	602.1 K (9.94 ×)
Res3D [23]	85.8%	33.2 (0.41 ×)	19.3 (4.60 ×)	301.1K (4.97 ×)
I3D [24]*	93.4%	25 (0.31 ×)	215.8 (51.38 ×)	16056.3 K (264.96 ×)
CoViAR [7]	90.3%	80.6 (1.00 ×)	4.2 (1.00 ×)	60.6 K (1.00 ×)
Proposed	88.5%	33.6 (0.42 ×)	3.5 (0.83 ×)	60.6 K (1.00 ×)

4.3.4. Comparison with the state-of-the-art methods

Our method is most efficient compared with state-of-the-art models in Table 10. Since for CoViAR and our model the P-frame and I-frame computational costs are different, we report the average GFLOPs over all frames. As shown in the table, our model is 1.2 times faster and 2.4 times smaller than CoViAR. I3D [24] has least number of parameters while being slowest due to excessive input data size. The calculation of input data size is shown in Table 11. Figs. 16 and 17 summarizes the results. Our model achieves the best efficiency and has few number of parameters, while having far smaller input data size. Note that some of the state-of-the-art methods in Table 10 can achieve higher accuracy

Table 11
Calculation of input data size.

	Calculation of input data size
Two-stream	224*224*3 + 224*224*20
C3D	16*112*112*3
Res3D	8*112*112*3
I3D	64*224*224*3 + 64*224*224*2
ResNet-152	224*224*3
CoViAR	224*224*(3+2*11+3*11) / 12

Table 12
The training time between CoViAR [7] and our proposed method. (hours).

	I-frame image	P-frame motion vector	P-frame residual	Total time
CoViAR [7]	71.2	10.42	10.53	92.15
Our proposed method	99.85	86.91	87.18	273.94
Difference time	28.65	76.49	76.65	181.79

with large-scale video datasets. For fair comparison, we used the accuracies only trained on the UCF-101 dataset.

Table 12 compares the training time between CoViAR [7] and our proposed method. Integrating the knowledge from multiple CNNs and distillating the knowledge out to student model make the training time longer than CoViAR. However, our application is mainly for inference, and for inference time, our method is more accurate and faster.

5. Conclusion

In this thesis, we compress the model which is the most efficient method for action recognition currently and improve the overall speed by using knowledge distillation technology to transfer its knowledge to a small model. The small model has richer knowledge than vanilla small model yet own less parameters and complexity than original cumbersome models. We also propose a multi-teacher knowledge distillation framework for compressed video action recognition to improve the accuracy after compression. We integrate the knowledge from different teachers, the comprehensive knowledge can promote the performance of the student. We explored multi-teacher knowledge distillation with various combinations of different teachers to further observe its impact. Experiments show that we can reach a 2.4 × compression rate and 1.2 × computation reduction with about 1.79% loss of accuracy on the UCF-101 dataset and 0.35% loss of accuracy on the HMDB51 dataset. Our approach achieves the best efficiency and has few number of parameters, while having far smaller input data size.

Declaration of Competing Interest

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

References

[1] O. Boujelben, B. Mohammed, Efficient fpga-based architecture of an automatic wheeze detector using a combination of mfcc and svm algorithms, J. Syst. Archit. 88 (2018) 54–64.
[2] B. Li, L. Ying, Z. Simin, Multi-task learning for intrusion detection on web logs, J. Syst. Architect. 45 (2017) 92–100.
[3] D. Madroñal, et al., Svm-based real-time hyperspectral image classifier on a many-core architecture, J. Syst. Architect. 80 (2017) 30–40.
[4] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.
[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.
[6] B. Zhang, L. Wang, Z. Wang, Y. Qiao, H. Wang, Real-time action recognition with enhanced motion vector CNNs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2718–2726.
[7] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A.J. Smola, P. Krähenbühl, Compressed video action recognition, CVPR, 2018.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 1, IEEE, 2005, pp. 886–893.
[9] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
[10] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked fisher vectors, in: European Conference on Computer Vision, Springer, 2014, pp. 581–595.
[11] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79.
[12] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 3551–3558.
[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.
[14] Z. Lan, Y. Zhu, A.G. Hauptmann, S. Newsam, Deep local video feature for action recognition, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on, IEEE, 2017, pp. 1219–1225.
[15] J. Zhu, W. Zou, Z. Zhu, End-to-end video-level representation learning for action recognition, arXiv preprint arXiv:1711.04161(2017).
[16] A. Diba, V. Sharma, L. Van Gool, Deep temporal linear encoding networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017.
[17] C. Feichtenhofer, A. Pinz, R.P. Wildes, Spatiotemporal multiplier networks for video action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 7445–7454.
[18] Y. Wang, M. Long, J. Wang, P.S. Yu, Spatiotemporal pyramid network for video action recognition, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2017.
[19] R. Girdhar, D. Ramanan, Attentional pooling for action recognition, in: Advances in Neural Information Processing Systems, 2017, pp. 34–45.
[20] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, B. Russell, Actionvlad: Learning spatio-temporal aggregation for action classification, in: CVPR, 2, 2017, p. 3.
[21] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[22] C. Feichtenhofer, A. Pinz, R. Wildes, Spatiotemporal residual networks for video action recognition, in: Advances in neural information processing systems, 2016, pp. 3468–3476.
[23] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri, Convnet architecture search for spatiotemporal feature learning, arXiv preprint arXiv:1708.05038(2017).
[24] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE, 2017, pp. 4724–4733.
[25] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, in: Advances in neural information processing systems, 2015, pp. 1135–1143.
[26] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient convnets, arXiv preprint arXiv:1608.08710(2016).
[27] H. Hu, R. Peng, Y.-W. Tai, C.-K. Tang, Network trimming: a data-driven neuron pruning approach towards efficient deep architectures, arXiv preprint arXiv:1607.03250(2016).
[28] J.-H. Luo, J. Wu, W. Lin, Thinet: a filter level pruning method for deep neural network compression, arXiv preprint arXiv:1707.06342(2017).
[29] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, in: Advances in neural information processing systems, 2014, pp. 1269–1277.
[30] X. Zhang, J. Zou, K. He, J. Sun, Accelerating very deep convolutional networks for classification and detection, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 1943–1955.
[31] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, arXiv preprint arXiv:1511.06530(2015).
[32] X. Yu, T. Liu, X. Wang, D. Tao, On compressing deep models by low rank and sparse decomposition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7370–7379.
[33] W. Chen, J. Wilson, S. Tyree, K. Weinberger, Y. Chen, Compressing neural networks with the hashing trick, in: International Conference on Machine Learning, 2015, pp. 2285–2294.
[34] J. Wu, C. Leng, Y. Wang, Q. Hu, J. Cheng, Quantized convolutional neural networks for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4820–4828.
[35] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531(2015).
[36] D. Lopez-Paz, L. Bottou, B. Schölkopf, V. Vapnik, Unifying distillation and privileged information, arXiv preprint arXiv:1511.03643(2015).
[37] J. Yim, D. Joo, J. Bae, J. Kim, A gift from knowledge distillation: fast optimization, network minimization and transfer learning, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2, 2017.
[38] W.M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, R. Pascanu, Sobolev training for neural networks, in: Advances in Neural Information Processing Systems, 2017, pp. 4278–4287.
[39] Z. Xu, Y.-C. Hsu, J. Huang, Learning loss for knowledge distillation with conditional adversarial networks, arXiv preprint arXiv:1709.00513(2017).
[40] A. Mishra, D. Marr, Apprentice: using knowledge distillation techniques to improve low-precision network accuracy, arXiv preprint arXiv:1711.05852(2017).
[41] A. Polino, R. Pascanu, D. Alistarh, Model compression via distillation and quantization, arXiv preprint arXiv:1802.05668(2018).

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, Ieee, 2009, pp. 248–255.
[43] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch(2017).
[44] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980(2014).
[45] K. Soomro, A.R. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402(2012).
[46] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2556–2563.
[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

Meng-Chieh Wu received B.S. degree in Electronic Engineering from National Ilan University in 2013, and M.S. degree in Computer Science from National Tsing Hua University in 2018.Her research interest focuses on image processing, computer vision, and machine learning.

Ching-Te Chiu received her B.S. and M.S. degrees from National Taiwan University, Taipei, Taiwan. She received her Ph.D. degree from University of Maryland, College Park, Maryland, USA, all in electrical engineering. She was an Associate Professor with National Chung Cheng University, ChiaYi, Taiwan. She was member of technical staff with AT & T, Murry Hill, New Jersey, and at Lucent Technologies, Murry Hill, New Jersey, and with Agere Systems. She is currently a Professor at the Computer Science Department and Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan. Her research interests include High Dynamic Range Image and Video Processing, Super Resolution, Pattern Recognition, High Speed SerDes design, Multi-chip Interconnect, and Fault Tolerance for Network-on-Chip design. Dr. Chiu won the first prize award, the best advisor award, and the best innovation award of the Golden Silicon Award in 2006. She serves as a TC member of the IEEE Circuits and Systems Society, Nanoelectronics and Gigascale Systems Group, and the IEEE Signal Processing Society, Design and Implementation of Signal Processing Systems group. She is the program chair of the first IEEE Signal Processing Society Summer School at Hsinchu, Taiwan 2011 and technical program chair of IEEE workshop on signal processing system (SiPS) 2013. She served as associate editor of IEEE Transactions on Circuits and Systems I and currently serves as associate editor of IEEE Signal Processing Magazine and Journal of Signal Processing Systems.