A General Dynamic Knowledge Distillation Method for Visual Analytics

Zhigang Tu, *Member, IEEE,* Xiangjian Liu, and Xuan Xiao

***Abstract*—Existing knowledge distillation (KD) method normally fixes the weight of the teacher network, and uses the knowledge from the teacher network to guide the training of the student network no-ninteractively, thus it is called static knowledge distillation (SKD). SKD is widely used in model compression on the homologous data and knowledge transfer on the heterogeneous data. However, the teacher network that with fixed-weight constrains the student network to learn knowledge from it. It is worth expecting that the teacher network itself can be continuously optimized to promote the learning ability of the student network dynamically. To overcome this limitation, we propose a novel dynamic knowledge distillation (DKD) method, in which the teacher network and the student network can learn from each other interactively. Importantly, we analyzed the effectiveness of DKD mathematically (see Eq. 4), and addressed one crucial issue caused by the continuous change of the teacher network in the dynamic distillation process via designing a valid loss function. We verified the practicality of our DKD by extensive experiments on various visual tasks, *e.g.* for model compression, we conducted experiments on image classification and object detection. For knowledge transfer, video-based human action recognition is chosen for analysis. The experimental results on benchmark datasets (*i.e.* ILSVRC2012, COCO2017, HMDB51, UCF101) demonstrated that the proposed DKD is valid to improve the performance of these visual tasks for a large margin. The source code is publicly available online at[1]**

***Index Terms*—Static Knowledge Distillation (SKD), Fixed-weight, Dynamic Knowledge Distillation (DKD), Continuous optimization, Model Compression, Knowledge Transfer**

I. Introduction

Knowledge distillation (KD) is an important technique since it can directly transfer knowledge from complex models to a simple model. For the traditional KD approaches, the teacher network is often fixed for knowledge transferring, thus they can be treated as static knowledge distillation (SKD). In the past few decades, SKD has attracted long-term research attention [1], [2] due to its wide range of applications in face recognition [3], video action recognition [4], image super resolution [5], *etc*. For most visual tasks, the SKD technology can optimize the basic network architecture to enable it to be applied in a smaller and more flexible way. Many studies have been performed to improve the SKD technology, which include the modifications of the loss function of SKD [6], the model structure of SKD [7], and the knowledge content extraction of SKD [8], but almost none of them take into account the drawback of SKD that the teacher network with fixed-weight constrains the network to learn more potential Knowledge. Typically, for the traditional SKD method, the teacher network is fully pre-trained, and the highly abstract semantic information learned from the teacher network is output to guide the optimization of the student network. This strategy can help the student network to quickly obtain the knowledge of the teacher network, but cannot explore the teacher model's potential comprehensively. In addition, the traditional SKD transfers knowledge from the teacher network to the student network, while they overlook an important fact that the teacher network also limits the student network to learn knowledge. How to address the issue of releasing restriction imposed by the teacher network is of great concern for SKD.

Issue 1 – Optimization of KD: The concept of KD was first proposed by Hinton *et al*. [2]. He set the output knowledge from the fixed-weight teacher network as soft labels, and allowed the student network to learn the real labels and the soft labels simultaneously. In this way, the knowledge can be transferred to the student network implicitly. Subsequently, Xie *et al*. [9] took the influence of the data itself into account and added noise to the data to enhance the generalization ability of the model. Komodakis *et al*. [10] considered to optimize the above mentioned SKD structure from the perspective of the feature layer. However, none of these fixed-weight methods break through the limitation of the teacher network itself, where the limitation is reflected in the fact that the amount of knowledge from a fixed-weight teacher network is settled. In summary, if the teacher network can be optimized dynamically, the successive updated knowledge of the teacher network can promote the optimization of the student network interactively.

Issue 2 – Assessment of KD: In SKD, there are many loss functions used to measure the effect of knowledge distillation. To assess the numerical similarity between feature layers of the teacher and the student, Mean Squared Error (MSE) [11] is a popular choice. It can enforce a strong constraint, which brings the outputs of the teacher network and the student network closer. To evaluate the distribution similarity of the feature layers, Maximum Mean Discrepancy (MMD) [12] is a suitable selection, which can pay attention to the similarity between layers from a higher dimension. To measure the similarity of the output probability, the softmax loss [7] is modified to allow the student network to learn the probability distribution of the teacher network. Besides, Kullback-Leibler

Zhigang Tu and Xiangjian Liu are co-first authors. Corresponding authors: Xiangjian Liu (liuxj96@whu.edu.cn) and Xuan Xiao (xiaoxuan1111@163.com).

Zhigang Tu and Xiangjian Liu are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 430079 Wuhan, China.

Xuan Xiao is with Renmin Hospital of Wuhan University, Wuhan 430060, China.

Manuscript received Aug.20, 2022.

[1]https://github.com/errllxj/DKD

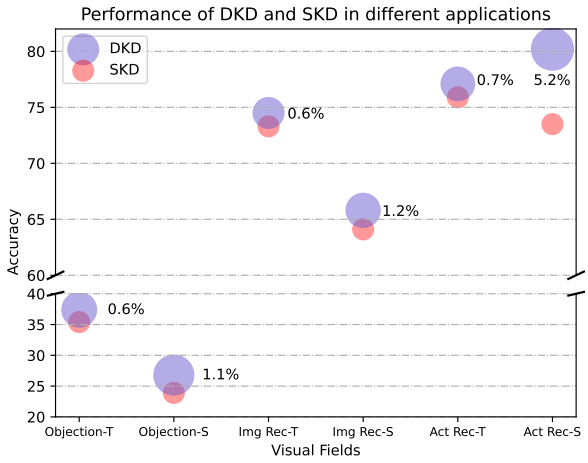

Fig. 1: Performance comparison of the proposed DKD and the traditional SKD in different visual application fields. "T" and "S" denote the teacher network and the student network, respectively. "Objection", "Img Rec", and "Act Rec" denote object detection, image classification, and video-based human action recognition, separately. Specifically, on these visual tasks, the accuracy comparison between our DKD and SKD is reported separately with respect to the teacher network and the student network, because our DKD can improve the accuracy of both, while SKD can only modify the accuracy of the student network. The percentage numbers represent the accuracy improvement of DKD over SKD. Taking 0.6% as an example, it represents that in the "Detection" task, the accuracy of the teacher network with using the proposed DKD is 0.6% higher than that of using the SKD.

Divergence (KLD) [13] can perform knowledge distillation of the output probability, but ignoring the information of the feature map. In general, the above loss functions can transfer knowledge from the teacher network to the student network from multiple perspectives. However, when the teacher network is optimized to improve the overall ability of KD, none of them can ensure to converge to the good result, causing the effectiveness of KD is non-guaranteed. In addition to the awareness of the limitations of the above-mentioned common loss functions, we also get inspiration by works from other areas. In GAN [14], since the Earth Mover's Distance (EMD) can well measure knowledge transfer between two features, and it is complementary to KLD, we therefore combine KLD and EMD into our final loss function to distill knowledge both from feature and label jointly. Besides, we analyze the availability of the combination "$KLD + EMD$" in mathematics (see Eq. 4 and Eq. 2) and by experiments (see IV-C1), which show that "$KLD+EMD$" avoids the common deterioration of the teacher network and the student network.

Issue 3.1 – Limitation of KD for model compression on homologous data: As a pioneer, Yim *et al*. [15] used the teacher network with frozen weights and designed a mutual distillation network model, but this method cannot well satisfy the requirement of model compression. It improves the robustness of the network, but the accuracy is relatively low. Later, Romero *et al*. [1] proposed a staged distillation strategy, which can deeply exploit the capability of the student network, and utilizes the teacher network to guide the student network for further optimization. However, this method only explores the student network unilaterally. Zhu *et al*. [16] tried to introduce the knowledge of the student network into the teacher network, and applied the gradient information of the teacher network to guide the training progress of the student network in turn, but this method makes no attempt to optimize the teacher network. In addition, for different tasks, Zagoruyko *et al*. [10] presented an attention map to optimize the KD in image classification. Wang *et al*. [17] designed a model to optimize the negative samples for KD in object detection. In summary, these SKD methods for model compression only improve the student network, neglecting the importance of the teacher network in the optimization process, which causing the student network to be unable to learn independently. Besides, the existing methods are not universal as they usually dedicated to make the KD model serves more suitable for a specific application.

Issue 3.2 – Limitation of KD for knowledge transfer on heterogeneous data: The main investigation in this area is to learn different kinds of knowledge across domains. Chen *et al*. [18] proposed a multi-modal KD method, with the usage of audio-visual knowledge to improve video representation learning. However, this method focuses on solving the problem of multi-source feature fusion, rather than studying how to integrate multi-source data. Wang *et al*. [19] exploited a multi-modal learning framework to migrate multiple teacher networks' information on the heterogeneous data to a student network. In summary, for these methods, mutual integration and promotion on the heterogeneous data are not well solved since the weights of the teacher network are fixed.

To address these issues of SKD, as shown in Figure 2, we exploit a dynamic knowledge distillation (DKD) method, which can continuously optimize the teacher network and the student network. Specifically, for our DKD, we integrate the knowledge of the student network to the teacher network to help exploring the potential of the teacher network. Besides, the teacher network will guide the optimization of the student network from multiple perspectives, e.g. feature learning, probability prediction. To test its effectiveness, we evaluate our DKD on two different applications (1) model compression on homologous data, including the visual tasks of image classification and object detection, and (2) knowledge transfer on heterogeneous data, including the visual task of video-based human action recognition.

For image classification, SKD mainly aims for compressing the model. By using our DKD method throughout the training process, the student network transfers part of the class probability knowledge to the teacher network. In turn, the teacher network guides the student network to optimize in terms of the feature map and the class probability. For object detection, based on the learned DKD in image classification, we contribute new modifications to extract the class probability, which will be described in detail in the subsequent section III-C. For video-based human action recognition, SKD primarily goals

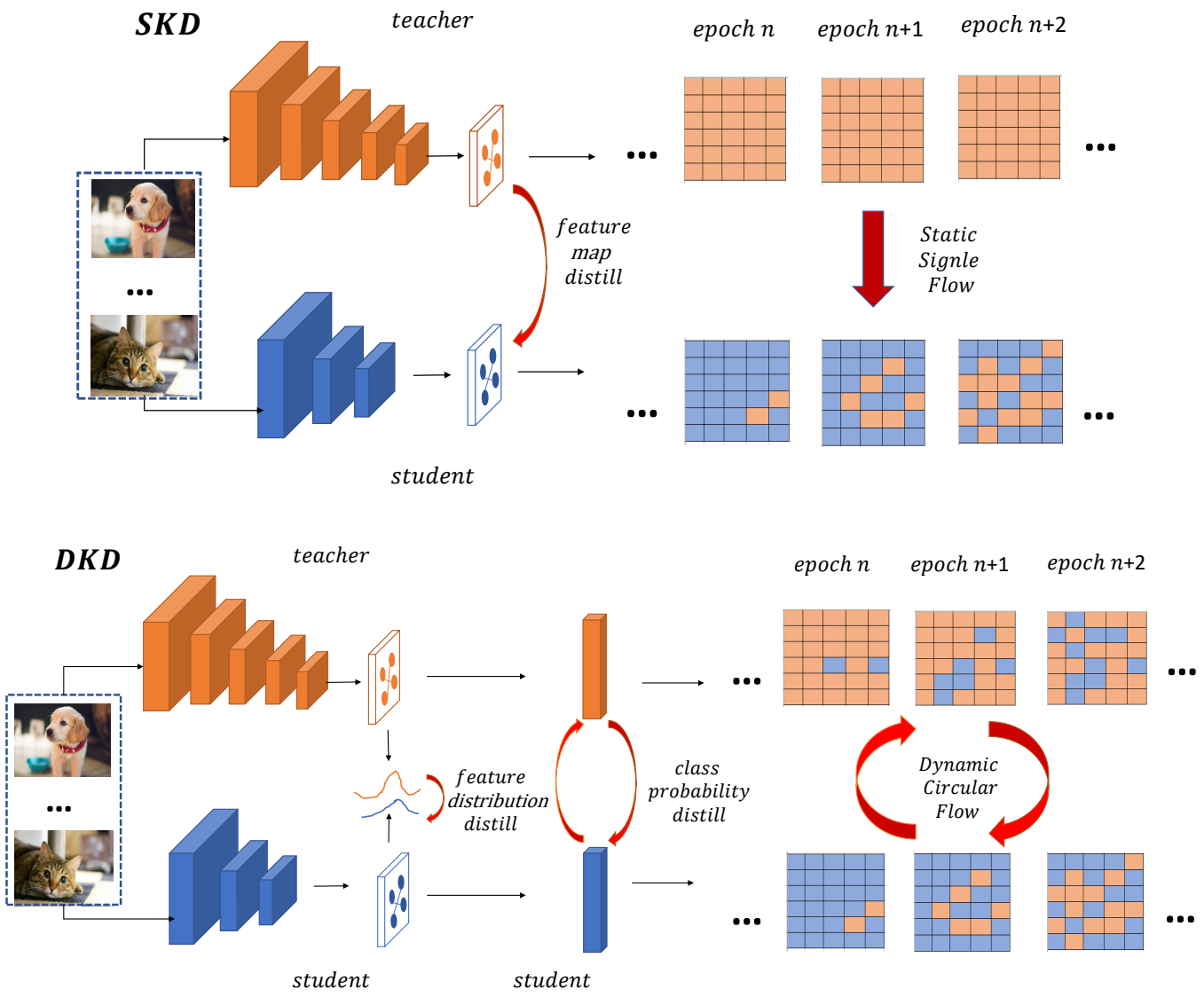

Fig. 2: Model difference between SKD and our DKD. For "SKD", the student network only distills the value of the feature map from the teacher network with fixed weights (*static*). In contrast, for our "DKD", the student network jointly distills both the distribution of the feature map (i.e. feature distribution) and the probability of the class prediction (i.e. class probability) from the teacher network with continuous-renewal weights (*dynamic*). Specifically, the teacher network and the student network are both dynamic, as they successively updating during the optimization process iteratively.

for knowledge transfer and knowledge distillation. Normally, this visual task needs to input two types of heterogeneous cues *i.e.* optical flow and RGB video [20], [21]. However, it is computational cost to estimate optical flow [22]. We use DKD to transfer the knowledge of optical flow to the RGB model. At the end, these two types of information can be obtained directly from the RGB model, accordingly, the efficiency is significantly boosted. The detail will be illustrated in the subsequent section III-D. In the experiment part IV-C1, we will indicate when it converges.

Our DKD greatly promotes the performance of SKD, and can be effectively applied in many domains. As shown in Figure 1, the accuracy of both the teacher network and the student network of DKD have been enhanced.

The main contributions are summarized in three fold:

- We propose a general dynamic knowledge distillation (DKD) method to enable the knowledge circularly updated between the student network and the teacher network during optimization. As a result, the knowledge of the student network and the teacher network can promote each other interactively, and their performance on different visual tasks are both constantly enhanced.
- An effective loss function, which combines EMD and KLD, is exploited to address the issue of teacher net-

works' knowledge change at the DKD optimization process. Remarkably, we prove DKD is convergent and analyze its effectiveness mathematically.
- We verify the explored DKD is useful for the applications of model compression on homologous data and knowledge transfer on heterogeneous data by testing it on various visual tasks, which including image classification, object detection and video-based human action recognition. Their results outperform SKD for a large margin.

II. Related Work

The proposed DKD method exploits a new way to distill knowledge dynamically and can be applied to many visual tasks, *e.g.* image classification, detection, video based human action recognition, etc. Comparing our DKD method to the conventional SKD method, great benefit can be gained. The related works about the investigation of KD on both the theory and application are introduced below.

A. Knowledge distillation (KD)

KD was first proposed to address the issue of model compression. To use a small model to obtain high performance, Hinton *et al*. [2] presented a teacher-student based KD framework, where a complex network is called the teacher network, and a simple network with lower accuracy is called the student network. In Hiton's design, the student network can extract the knowledge of the teacher network by learning the feature maps. Except to directly transmitting the network knowledge to the student, the teacher network can also teach the student network to learn online. Heo *et al*. [23] stated that the student network can apply the learned skills in different tasks to capture the boundary decision-making of the network. Zhang *et al*. [13] used the Kullback–Leibler divergence (KLD) to turn the class probability predicted from the teacher network into a label to allow the student network to learn independently. This method no longer distinguishes the difference between the teacher network and the student network, and shows that only the teacher network learns independently can bring great changes. Ahn *et al*. [24] proposed to increase the characteristics of the middle layer between the teacher network and the student network from the perspective of mutual information (MI). This method just considers how to better measure the results of KD from other aspects. Different from these methods, we explore a general DKD network, which can significantly improve the accuracy of the teacher network and drive the student network to obtain higher performance jointly.

B. Image classification with KD

Image classification is one typical visual application of KD for model compression. To better perform image recognition, deep learning has been foremost studied in this field. From the early VGG [25], GoogleNet [26] to the subsequent outstanding ResNet [27], the network structure for image classification has been improved progressively. However, these networks are too complicated to be applied to mobile devices. Many researchers have tried to lighten the deep learning model by KD. Yang *et al*. [28] changed the model structure to realize knowledge distillation, but he ignored the importance of the feature map information in image classification. Chen *et al*. [29] created a fast-to-execute student model to mimic a large teacher network, which can guarantee the transfer information of the feature layer. Guan *et al*. [30] introduced the feature aggregation strategy to imitate the multi-teacher distillation in the single-teacher distillation framework. He brought more information by adding more teacher networks. However, the types of the input information are limited, and adding more teacher networks with the similar input cannot fundamentally improve the learning ability of the student network. Besides, in the subsequent investigations, all the teacher networks fix their weights, which constraining the amount of knowledge that the student network can learn. In contrast, the method of DKD proposed by us breaks the limitations due to the teacher network, where it enables the student network to optimize continuously.

C. Object detection with KD

Object detection is another typical visual application of KD for model compression. The object detection technique has undergone rapid progress since many kinds of object detection algorithms have been exploited, such as RCNN [31], Fast RCNN [32], Faster RCNN [33], YOLO [34], FCOS [35] and so on. One phenomenon is that the object detection models become larger and larger, therefore it is urgent to light-weight them. KD provides a choice to address this issue. Chen *et al*. [36] used the traditional Faster RCNN as the basis to conduct KD in three aspects: the feature layer, the classification layer, and the regression layer. This method transfers the knowledge of a large teacher model to a small student model, and achieves high accuracy. Hoffman *et al*. [37] incorporated the depth information to train a RGB object detection model at the KD process. This approach improves the performance by introducing additional information in KD, which greatly limits its application in practice. Dai *et al*. [38] proposed a new distillation method for object detection based on discriminative examples, which does not consider the positive or negative distinctions of the ground truth. Wei *et al*. [39] presented a Quantization Mimic, which first quantizes the large network, then mimics a quantized small network. However, this method only concerns the information interaction from the feature maps. In object detection, the information that can be learned is diverse, such as regression box, prediction class, *etc*. However, the above object detection methods with KD, the knowledge flows in one direction. Different from them, With our DKD, the knowledge of the teacher network and the student network is formed into a circulation, and the accuracy of them are also improved simultaneously.

D. Video human action recognition with KD

Video recognition is an important visual application of KD for knowledge transfer. For the special video-based human action recognition, one of the representatively techniques is the two-stream network [40], in which one is the RGB stream that is used to extract the appearance information, and another

is the optical flow based motion stream that is applied to capture the motion information. Because the parameters of the two-stream model are too large, many methods, which aim to reduce the model complexity by KD, have been constantly proposed. Stroud *et al*. [41] improved the representation of motion through KD, which effectively merged the two streams into a single stream to reduce the complexity of the model. But this method also brings a decrease in accuracy. Considering optical flow estimation is time consuming and preventing the two-stream models to be applied to some real-time tasks, Crasto *et al*. [11] explored a method via KD to allow the RGB model to simulate the output of the optical flow stream. In this way, it only requires optical flow in the training phase while avoids to use optical flow in the testing phase. But in order to get better results, a multiple-stream fusion strategy is utilized, which significantly decreases the goal of lightening the model. On the other side, many 3D models play an important role in action recognition, e.g. I3D [42], C3D [43], etc. KD is also effectively used in 3D models. Feichtenhofer *et al*. [44] presented a SlowFast network, which reduces the amount of 3D network parameters, and uses KD to combine the motion and appearance information to simulate the human eye to observe moving objects. Different from the above methods, we apply the proposed DKD to the task of video-based human action recognition. Our DKD circles the knowledge of the optical flow model and the RGB model, so, the RGB model gets more motion information, and the optical flow model obtains more apparent information. Finally, the two are combined to obtain a better action recognition model.

III. Methodology

Our goal is to design an effective DKD framework, which can be applied in various visual fields for model compression (*e.g.* image classification and object detection) and knowledge transfer (*e.g.* video based human action recognition). Figure 2 shows the overall framework of our DKD model. Specifically, the teacher network will guide the student network to learn knowledge twice, *i.e.* in the feature map and in the class label, to transfer knowledge to the student network. Importantly, when the student gets the knowledge of the teacher, the student will also transmit knowledge to the teacher network, making the teacher network continuously improves itself.

A. The structure of DKD

Figure 3 shows the diagram of the proposed DKD model. The teacher network is large and has vast parameters, while the student network is much smaller. We use different distillation functions (*i.e.* EMD and KLD) at the two stages to extract the feature knowledge and the label knowledge.

In the first stage, the feature map of the student network (we denote it as X here), which is learned to imitate the feature map of the teacher network (we denote it as Y here) by utilizing the EMD. The $EMD(X,Y)$ metric with regard to the feature map distributions of X and Y over $\chi \times \chi$ can be defined as [45]:

$$EMD[X,Y] = \frac{inf}{\gamma \subset \Gamma} \int_{\chi \times \chi} \gamma ||x-y||_2 d_\gamma(x,y) \quad (1)$$

where Γ is the set of all possible joints on $\chi \times \chi$.

$EMD(X,Y)$ measures the minimum cost that changes from the distribution of feature map X to the distribution of feature map Y [45]. The movement from X to Y is done via the coupling γ. It is easy to see that when the difference between the probability distributions of X and Y is getting closer and closer, the $EMD(X,Y)$ is approaching to 0. An in-depth explanation of these concepts can be found in [46]. Due to this characteristic, compared with the traditional Mean Squared Error (MSE) and Maximum Mean Discrepancy (MMD), $EMD(X,Y)$ has a loose constraint relationship and is suitable for measuring the difference between the feature map distributions of the student network and the teacher network (*i.e.* X and Y here) in the dynamic model.

In the second stage, KLD is used as the constraint loss, which is defined as follows:

$$KLD(l_t, l_s) = \sum_{i=1}^{N} \sum_{m=1}^{M} l_t^m(x_t) \log \frac{l_t^m(x_t)}{l_s^m(x_s)}. \quad (2)$$

As shown in Figure 3, $l_t^m(x_t)$ and $l_s^m(x_s)$ refer to the probability of class m at sample x. KLD can describe the distance between the two probability distributions, so it is more suitable for distilling the class probability. l_t and l_s here are called "soft labels". Compared with the real labels which only have two values of 0 and 1, the information in the "soft labels" is richer. We input the "soft labels" of the student network as the additional knowledge to the teacher network. Although the accuracy of the "soft labels" from the student network is low, the wrong information is also valuable and can help the teacher network to learn more error examples. Through the above two stages of distillation, the joint optimization of the student network and the teacher network is realized. $LossAT(l,Y)$ is task-related, which represents the gap between the output label and the real label, and it will change accompanying with different tasks.

The overall loss function for our DKD is formulated as follows:

$$\begin{aligned} Loss_T &= \beta_1 KLD(l_s, l_t) + LossAT(l_t, Y) \\ Loss_S &= \alpha EMD + \beta_2 KLD(l_t, l_s) + LossAT(l_s, Y) \end{aligned} \quad (3)$$

where α, β_1, β_2 are the zoom scales. For β_1 in the teacher network, it can limit the cost of learning "soft labels" from the student network. Excessive student network information will carry a large number of wrong labels, which will lead the teacher network to deterioration in the wrong way. An appropriate β_1 can control the optimization of the teacher network. $Loss_T$ and $Loss_S$ denote the loss function of the teacher network and the student network, respectively. $\alpha EMD + \beta_2 KLD$ can form a dynamic flow in which the student network learns knowledge from the teacher network, and $\beta_1 KLD$ can form a dynamic flow in which the teacher uses the student's knowledge to optimize itself. Finally, incorporated $\alpha EMD + \beta_2 KLD$ with $\beta_1 KLD$, a dynamic circulation flow is formed in our DKD model.

The DKD model we proposed can distill knowledge from the feature layer as well as the final class probability. For most visual tasks, both types of information are available, thus our

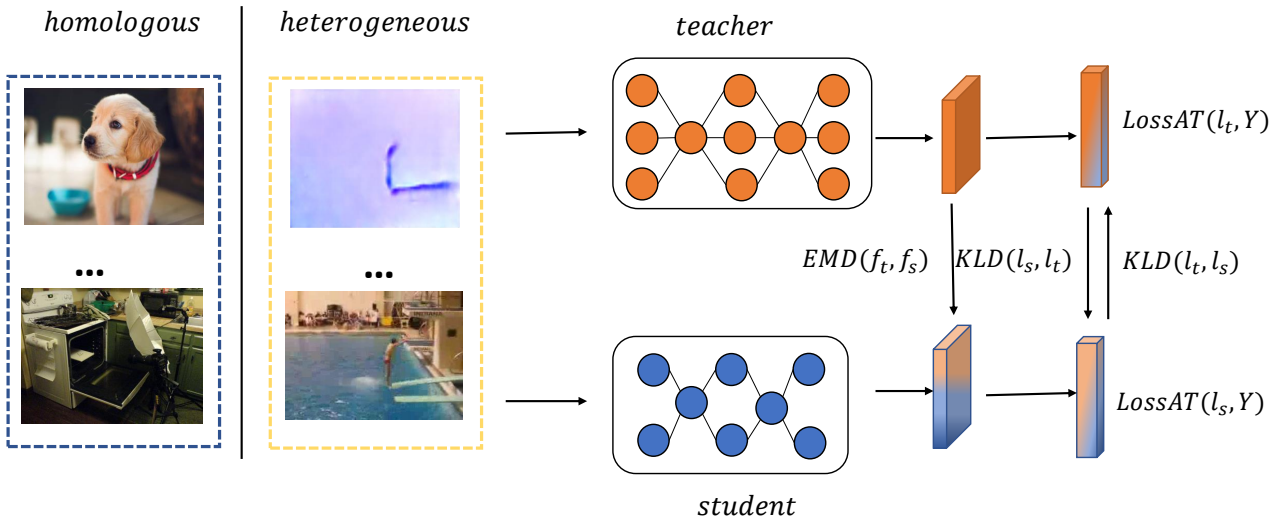

Fig. 3: The diagram of our DKD model. It includes both the feature distribution distillation and class probability distillation. In the feature distribution distillation stage, we use EMD as the constraint function, and in the class probability stage, we use KLD for mutual distillation. For model compression, the teacher network and the student network input the homologous data. For knowledge transfer, the two networks are inputted by the heterogeneous data. Notably, our DKD model is easy to design, in addition to KLD and EMD, $LossAT(l_t, Y)/LossAT(l_s, Y)$ (the subscript "s" and "t" respectively refers to "student" and "teacher") represents a task-related loss function which changes accompanying with the visual tasks.

DKD can be easily applied to them and without too much variation. It means that our DKD is an universality method.

B. The advantage of EMD

From Eq. 3, we see that the core of our DKD is that it can effectively distill knowledge in a dynamic way, however, the continuously optimized teacher network in DKD will cause non-convergence to KD. To solve this problem, we chose EMD as the loss function. Remarkably, we analyze its effectiveness mathematically below. Besides, we also verify its performance in the dynamic distillation experiments (see subsection IV-C2).

We set $D(x, y)$ as the distance metric. For the EMD, we can directly get the following mathematical formula from [45]:

$$\begin{aligned} D(x, y + \triangle y) &\leq D(x, y) + D(y, y + \triangle y) \\ &\leq D(x, y) + V^{\frac{1}{2}} \end{aligned} \tag{4}$$

where $V = \mathbb{E}[||\triangle y||_2^2]$ is the variance of $\triangle y$. When the teacher network is optimized based on the pre-training weights, we set a very small learning rate to it, which is equivalent to add a small change in y to become $y + \triangle y$. $D(x, y + \triangle y)$ is only related to $\triangle y$ except for x and y. In addition, there is an upper limit for $D(x, y + \triangle y)$. Consequently, the network with EMD is convergence.

In contrast, for MSE, we can get the following formula:

$$D(x, y + \triangle y) = D(x, y)^2 - 2 \triangle y D(x, y) + \triangle y^2 \tag{5}$$

According to this formula, we find that the $D(x, y + \triangle y)$ for MSE is related to the multiplication of $\triangle y$ and $D(x, y)$. Compared with EMD, $\triangle y$ has an additional scaling scale related to the original data $D(x, y)$. If the distribution gap between x and y is too large or too small, the small $\triangle y$ will not work, therefore, the network with MSE is difficult to converge.

From the above derivation process, we can summarize the following advantages of EMD: (1) In the dynamic networks, the EMD is just related to $\triangle y$ (derives from the learning information) and has an upper limit, which enables the entire network to converge. In contrast, MSE used in static distillation does not meet this requirement; (2) According to [47], we know that EMD can continuously transform one distribution into another while maintaining the geometric characteristics of the distribution itself. Which means the student network can maintain its independence while learning the knowledge of the teacher network. This advantage ensures the student network changes slowly following with the teacher network's variation; (3) The generally used functions, e.g. Kullback-Leibler (KL) divergence [48] and Jensen-Shannon (JS) divergence [48], can measure the probability distribution. However, when the probability distributions of the teacher network and the student network overlap little, they fail to assess the two distributions of these two networks accurately. In contrast, the EMD can reflect the distance of the two distributions precisely, since the student network is trained from scratch, its probability distribution overlaps very little with the teacher network. Due to these reasons, EMD is suitable for measuring the distance between the teacher network and the student network in the dynamic model.

C. Model compressing on homologous data

1) Image classification: Figure 4 shows the proposed DKD method for image classification, and we choose ResNet as the basic network for experiments due to ResNet is one of the

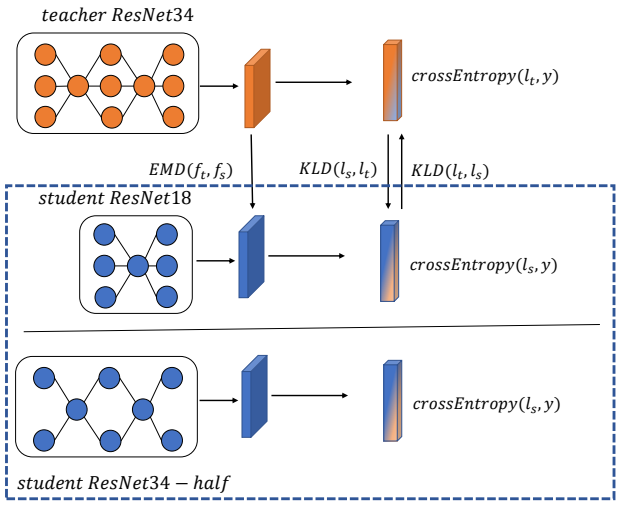

Fig. 4: The diagram of our DKD based image classification model (ResNet is chosen as the basic network for example). We show two kinds of student networks: where ResNet34 is selected as the teacher network, one student network is ResNet18, and the other student network is ResNet34-half.

most widely used networks in computer vision. As shown in the figure, we divided DKD into two steps. Specifically, in the first step, the student distills the feature map by EMD; in the second step, the student and the teacher mutually distill the class probability by KLD. *$Loss(l, Y)$ in Eq. 3 at here refers to the cross-entropy loss function*, accordingly, the overall formulation of our DKD based image recognition is expressed as:

$$\begin{aligned} Loss_{imaT} &= \beta_1 KLD(l_s, l_t) + CrossEntropy \\ Loss_{imaS} &= \alpha EMD + \beta_2 KLD(l_t, l_s) + CrossEntropy \end{aligned} \tag{6}$$

To better realize DKD, we designed two ways to combine the teacher network and the student network. As shown in Figure 4, the first one is that both the teacher network and the student network are completely same as the original ResNet model, e.g. ResNet34 (teacher) and ResNet18 (student). The second one is that the student network derives from the teacher network, e.g. ResNet34 (teacher) and ResNet34-half (student). Here, ResNet34-half represents to halve the number of channels of ResNet34. The reason for choosing these two student networks is that, compared with the teacher network, ResNet18 is mainly different in the number of network layers, and ResNet34-half is different in the network channel.

2) Object detection: Figure 5 shows the application of our method in object detection. We select the FCOS network [35] as the baseline for experimenting. The overall knowledge distillation framework is same as our basic DKD. For the pyramid cascade structure in FCOS, we use KLD to distill each layer. *The $Loss(l, Y)$ in Eq. 3 at here refers to the original FCOS loss function*, which includes a cls_{loss} for class prediction, a reg_{loss} for box regression, and a $centerness_{loss}$ for centerness. The overall formulation of our DKD based object detection is expressed as:

$$\begin{aligned} Loss_{detT} &= \beta_1 \sum_{i=1}^{n} KLD(l_s, l_t) + Loss_{FcosT} \\ Loss_{detS} &= \alpha EMD + \beta_2 \sum_{i=1}^{n} KLD(l_t, l_s) + Loss_{FcosS} \end{aligned} \tag{7}$$

where n represents the level of the pyramid (we set $n = 5$ as [35]). We perform DKD based mutual distillation on all 5 layers. Since different layers have different sizes in object detection, the DKD at all layers can learn useful knowledge to the greatest extent.

D. Knowledge transfer on heterogeneous data

1) Video-based human action recognition: In addition to compressing model, knowledge distillation also has many applications in knowledge transfer on heterogeneous data. Importantly, video-based human action recognition is a representative task. In the field of human action recognition, the two-stream network is one of the primary strategies [40]. The first stream is aiming to learn the motion information mainly from optical flow, and the second stream is used to capture the appearance information from RGB. However, it is time consuming to extract the optical flow information, which hinders the practical application of the two-stream method. Although optical flow and RGB are two types of heterogeneous data, considering that optical flow is derived from RGB video frames, it is feasible to migrate the optical flow information to the RGB model. In the SKD based method [11], as shown on the left side of Figure 6, optical flow is set as the teacher network, RGB is set as the student network, and MSE is used as the constraint. However, this method has similar drawbacks like SKD, *i.e.* the teacher network is static. To address this issue, we apply our DKD to this network (see the right side of Figure 6). The 3D ResNeXt-101 [49] is selected as the basic network for experimenting. Approximate to the above visual tasks, DKD here also distills two kinds of knowledge, *i.e.* the feature map and the class probability. *The $Loss(l, Y)$ in Eq. 3 at here refers to the cross-entropy loss function*. The overall formulation of our DKD based human action recognition is expressed as:

$$\begin{aligned} Loss_{actT} &= \beta_1 KLD(l_s, l_t) + CrossEntropy \\ Loss_{actS} &= \alpha EMD + \beta_2 KLD(l_t, l_s) + CrossEntropy \end{aligned} \tag{8}$$

Since the data source of the teacher network and the student network is different, it is easy to produce poor results if we start the DKD training directly from the scratch. Therefore, we firstly use the SKD to make the student network information which is learned from RGB and the teacher network information which is extracted from optical flow close to each other. Then, the proposed DKD is utilized to break through the limitations of SKD – enabling the RGB model to better learn and integrate the knowledge of the optical flow model. As a result, the performance of both the teacher network and the student network is improved.

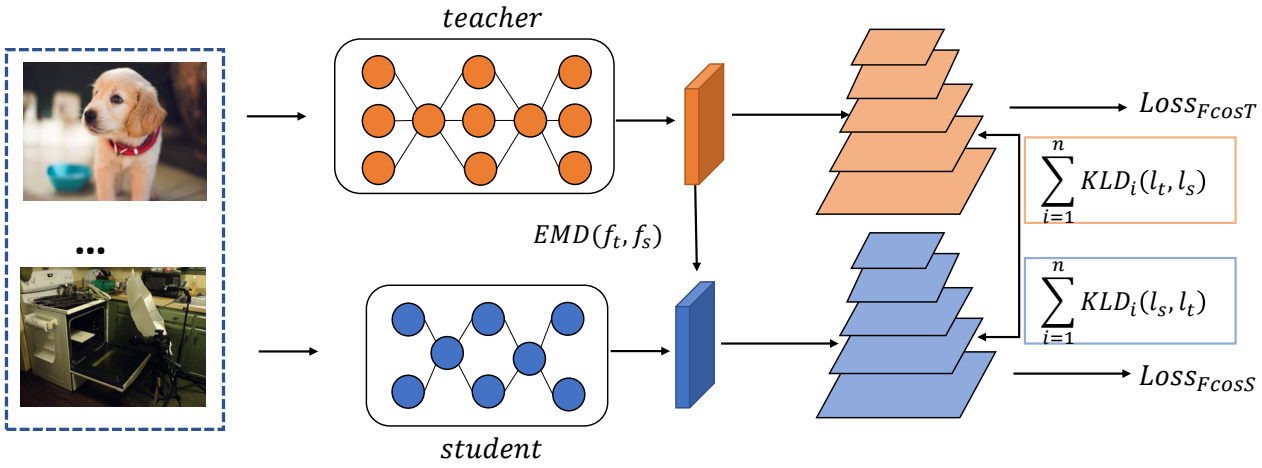

Fig. 5: The diagram of our DKD based object detection model. We use the FCOS model [35] to explore the effect of DKD in object detection. In this model, we construct a pyramid structure, which enables us to distill the class probability at each level. As well, KLD is chosen as the constraint loss function.

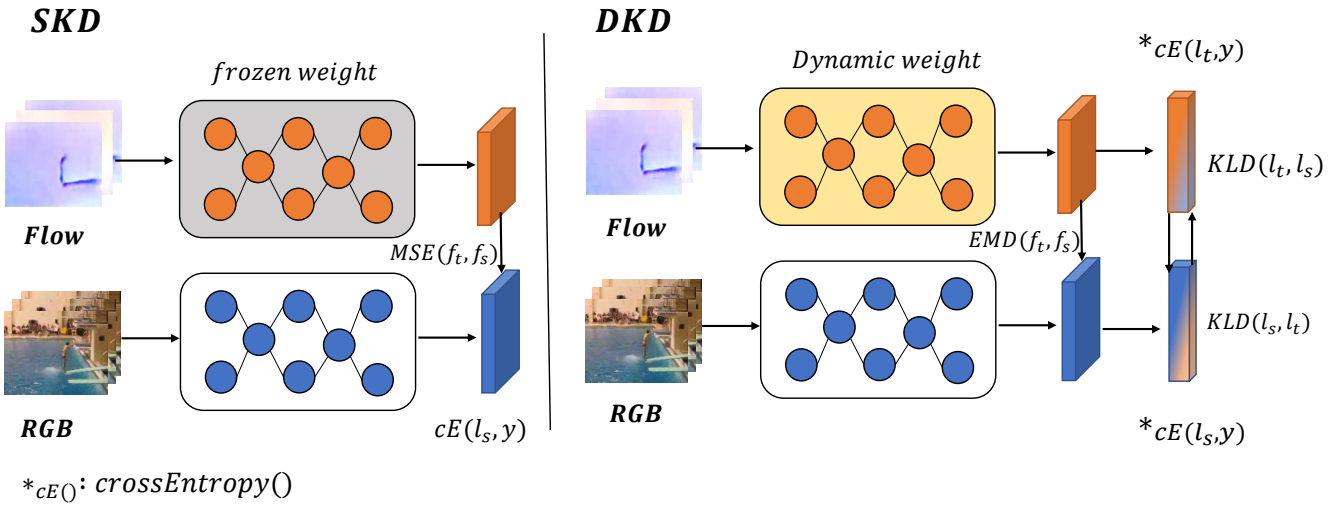

Fig. 6: The diagram of our DKD based action recognition model. "Flow" represents the optical flow model. We compare two kinds of knowledge distillation models, *i.e.* "SDK" and "DKD". In "SDK", the optical flow weights are freezed, and the optical flow based teacher network uses MSE as the constraint loss function. In "DKD", both the optical flow stream and the RGB stream are dynamic networks, respectively with EMD and KLD as the constraint loss function.

E. Summarization of DKD in different visual tasks

Our DKD shares the same spirit across different visal tasks, where the information of the teacher and student networks forms a dynamic cycle. In this cycle, the student network focuses on improving its feature extraction (EMD) and recognition (KLD) while the teacher network identifying wrong recognition (KLD). This property is generic for various visual tasks (*e.g.*, image recognition, object detection, video action recognition, etc.). Accordingly, when applying DKD to the tasks in different domains, $LossAT(l, Y)$ at Eq. 3 is the only part that requires targeted adjustment. E.g., for image recognition, we directly apply DKD and set $LossAT(l, Y)$ to $CrossEntropy$ loss at Eq. 6. For object detection, we utilize DKD multiple times in the pyramid structure and set $LossAT(l, Y)$ as the $Loss_{Fcos}$ loss at Eq. 7. For video action recognition, we change $LossAT(l, Y)$ to $CrossEntropy$ loss at Eq. 8.

IV. Experiment

We extensively validate the effectiveness of our DKD method on three different visual tasks with bench-

mark datasets: image classification using the ImageNet ILSVRC2012 [50] dataset, object detection using the MS COCO 2017 dataset [51], and video-based human action recognition using both the HMDB51 [52] and UCF101 [53] datasets. We then ablate the key ingredients of DKD to verify our design.

A. Evaluation Datasets

We conduct experiments on benchmark datasets of three visual tasks to evaluate the versatility of our DKD method.

ILSVRC2012. ILSVRC2012 is one of the ImageNet dataset family members, serving as a popular benchmark for large-scale image classification. Its images are collected through Flickr and other search engines. The training set consists of over 1.2 million images covering 1,000 semantic categories which include both the internal nodes and leaf nodes of ImageNet. The validation and testing sets consist of 50,000 and 100,000 images, respectively, where the later has no publicly available class annotations.

MS COCO2017. MS COCO2017 is a publicly accessible large object detection dataset produced and maintained by Microsoft. It covers 80 object categories with diverse scenes, and all its images are collected via Flickr and Amazon's Mechanical Turk (AMT). MS COCO2017 contains 118287, 5000, and over 40670 images for training, validation, and testing, with high quality annotations.

HMDB51. HMDB51 consists of 51 classes of actions covering 6849 videos collected from YouTube, and other video websites. Each action has at least 51 videos with the resolution of 320×240. Its action class includes general facial actions, facial manipulation and object manipulation, general body actions, interactive actions with objects, and human body actions.

UCF101. UCF101 provides videos of 101 classes of actions captured from YouTube. Each action class consists of moves of 25 people, where each person exhibits 4 to 7 groups of actions. It contains 6.5G video by total, where all videos are fixed to the resolution of 320×240. All 101 classes of actions can be gathered into 5 sets: actions of human-object interaction, human-human interaction, musical instrument performance, and sports action. UCF101 covers diverse object variations including changes of appearance, posture, object scale, background, texture, and *etc*.

B. Implementation Details

All experiments are implemented on Pytorch, the hyper-parameter settings of training/evaluation with respect to different visual tasks are adjusted accordingly.

Image classification. We compare our DKD method with other related methods on the raw and modified ResNets [27]. We adopt the basic training/evaluation protocols to randomly reshaped and cropped input images to the resolution of 224×224. Models are trained via SGD optimization with weight decay of 10^{-4} and momentum of 0.9. We assign different start learning rates for the teacher and student networks, where the former and later are set to 10^{-5} and 0.1 by default, respectively. Note that the teacher network usess the pre-trained weights while the student network trains from scratch.

Object detection. Our DKD model and the compared methods are evaluated on FCOS [35], which is a popular object detection framework. We adopt the Adam optimizer. The input images, which are originally 224×224, are reshaped to the resolution of 667×667 to adapt to our experiments. The learning rates for both the teacher and student networks are set to 10^{-4}. Note that the pre-trained weights of the teacher network are not trained exhaustively, we presume that there is still room for further improvement.

Video action recognition. We employ $3D$ ResNeXt-101 [49] as the backbone. For optical flows, we extract frames at 25 fps and resize them to at least 256 pixels for each image. We utilize the TV-L1 algorithm [54] to compute optical flow with default OpenCV parameter settings. As each optical flow is a two-dimension vector, we preserve u and v, *i.e.*, the corresponding directional components of x- and y-axis, in two different pictures, respectively. We truncate the values of all optical flows to a bounded range $[-20, 20]$ and then map them to $[0, 255]$. Following with [11], clips with 64 frames are selected as the inputs for networks. At the training phase, we randomly crop all input video frames to the resolution of 112×112. Each input video frame and optical flow image are subtracted by the ActivityNet [55] mean of the RGB frame and a value of 127.5, respectively. For the RGB and optical flow streams, we use the SGD optimization with a weight decay of 0.0005 and a momentum of 0.9 to fine-tune the models that are pre-trained on Kinetics400. At the testing phase, we apply center crop to all non-overlapping clips and calculate their average scores.

C. Model Compression on homologous data

TABLE I: The performance of ResNet18 with different training strategies. "From scratch + Dynamic" denotes training ResNet18 with our proposed DKD from scratch, and "From weight + Dynamic" represents training ResNet18 with our proposed DKD by loading the pre-trained weights from the public accessible model.

ResNet18	Accuracy
From scratch	64.7
From scratch+Dynamic	70.9
From weight	69.8
From weight+Dynamic	70.6

1) Image recognition: For image recognition, we set the parameters α , β_1, β_2 of our DKD (see Eq. 6) to 0.0025, 0.1 and 0.5, respectively, according to its experimental performance. We observe the loss value of each part, and find that when the overall loss converges to a minimum, each part, *i.e.* $\alpha \times EMD$, $\beta_2 \times KLD$ and $CrossEntropy$, has the similar value in the student network. In other words, when the model is optimal, the effect of each part is close to the same. To evaluate which training strategy is more suitable for the proposed DKD, we utilize the popular ResNet34 [27] and ResNet18 [27] as the backbone for the teacher network

TABLE II: Comparing the effects of our dynamic distillation (DKD) and the traditional static distillation (SKD) for image classification on the ILSVRC2012 dataset. Since SKD does not improve the accuracy of the teacher network, the accuracy of "SKD" and "Raw model" in the teacher network are the same.

Model	Raw model	SKD	DKD
ResNet34(t)	73.3	73.3	73.9
ResNet18(s)	69.7	70.3	70.9
ResNet152(t)	78.3	78.3	78.7
ResNet50(s)	76.1	76.5	76.8
ResNet34(t)	73.3	73.3	73.8
ResNet34-half(s)	64.1	64.7	65.3
ResNet50(t)	76.1	76.1	76.6
ResNet50-half(s)	71.7	71.9	72.2

and the student network. Note that the accuracy of ResNet18 in our experiment with the strategy "From scratch" is lower than "From weight", since no targeted training augmentations are applied. As shown in Table I, by using our dynamic distillation, no matter with ("From weight+Dynamic") or without ("From scratch+Dynamic") the pre-trained weights, clear gains on accuracy are achieved (70.6% vs 69.8%, 70.9% vs 64.7%). Specially, the strategy dynamic distillation with training from scratch (*i.e.* "From scratch+Dynamic") shows superior result to others. Moreover, Figure 7 further demonstrates that dynamic distillation with training from scratch (*i.e.* "From scratch+Dynamic") introduces both higher accuracy and faster optimization. This is because for dynamic distillation with pre-trained weights, it only shows notable improvement on accuracy at the late training stages, indicating that the dynamic distillation is insufficiently utilized when training with the pre-trained weights. It means that our DKD can help the network to improve the distillation effect in the initial and final stages, and avoiding the situation where both the teacher network and the student network become bad simultaneously.

As reported in Table II, the traditional SKD can only improve the accuracy of the student network, in contrast, our DKD introduces gains to both the teacher network and the student network. Furthermore, our DKD achieves much higher improvement to both the simple and complex networks than SKD. Specifically, for the model with ResNet34 as a teacher network and ResNet18 as a student network, our DKD gets a significant enhancement. Specifically, compared with "Raw model", SKD and DKD respectively introduces 0.6% (70.3% vs 69.7%) and 1.2% (70.9% vs 69.7%) increases on accuracy, where the improvement of DKD is double that of SKD. For the teacher-student of ResNet152 and ResNet50, SKD and DKD show 0.7% (76.8% vs 76.1%) and 0.4% (76.5% vs 76.1%) improvements on accuracy in contrast to "Raw model", where the improvement of DKD is 75.0% higher than that of SKD. Noticeably, the teacher-student model "ResNet152-ResNet50" contains far more parameters than "ResNet34-ResNet18", which means that the improvement brings by our DKD on the simple model is more significant.

To evaluate the case that the teacher-student networks under the same structure but with different parameters, we reduce the half number of filters on each layer of ResNet34 and ResNet50 to produce the networks ResNet34-half and ResNet50-half. We respectively utilize ResNet34-half and ResNet50-half as the student networks and use their corresponding original ResNets as the teacher networks. Table II shows that our DKD obtains accuracy improvement by 0.5% (73.8% vs 73.3%) on ResNet34 and 1.2% (65.3% vs 64.1%) on ResNet34-half, as well as 0.5% (76.6% vs 76.1%) on ResNet50 and 0.5% (72.2% vs 71.7%) on ResNet50-half. These results reveal that DKD boosts the corresponding student network more significantly than SKD, since the student network contains fewer learnable parameters (*i.e.*, lower capability of representation). Besides, DKD introduces superior enhancement to both learning strategies of extracting knowledge with scratch and SKD. In summary, DKD shows better performance to the models that with fewer learnable parameters.

We analyze the reason why DKD exceeds SKD and find that DKD forms a dynamic circular flow. The motivation of our dynamic circular flow can be summarized into two points: (1) From the perspective of the theoretical point, dynamic circular flow can form a positive feedback between the teacher network and the student network. With EMD, the knowledge of the student network is used to form a small interference to the teacher network like $\triangle y$ in Eq. 4, which promotes the optimization of the teacher network. Meanwhile, the student network can further learn knowledge from the teacher network by KLD. In this way, a dynamic circulating flow is formed, which can bring positive changes to the teacher network and the student network simultaneously. (2) From the perspective of model potential, the dynamic circulation flow can break through the limitations brought by the teacher network with fixed weights to explore the model potential. Specifically, the network with SKD is unable to adjust its output due to the weights cannot be interactive renewed continuously. In contrast, the proposed DKD is able to assist the teacher network to avoid incorrect information guided by the interactive knowledge of the student network, which helps the network to overcome the constraint of itself.

2) Object detection: To verify the effectiveness of our DKD method on the task of object detection, two popular frameworks, *i.e.* RetinaNet [56] and FCOS [35], are selected for evaluation on different accuracy metrics. All input images are resized to the resolution of 667×667 pixels. We train all methods from scratch.

For RetinaNet, we respectively apply RetinaNet-ResNet50 (DKD) and RetinaNet-ResNet50-half (DKD) as the teacher network and the student network. Table III reports the performances of RetinaNets with and without our DKD. Models with DKD outperforms their counterparts without DKD by 1.0% (30.8% vs 29.8%) and 0.8% (28.3% vs 27.5%) in accuracy on AP (*i.e.* the primary metric), respectively. Particularly, the student network helps the teacher network achieves significant accuracy improvement in detecting medium-size and large-size objects, where the gains can reach to AP_m 1.6% (35.1% vs 33.5%) and AP_l 1.5% (45.4% vs 43.9%), respectively. Meanwhile, the teacher network helps the student network in detecting small-size objects to get improvement by AP_s 0.9% (11.3% vs 10.4%) as the positive feedback. The good object

TABLE III: Experiment on RetinaNet [56] for object detection on the MS COCO2017 dataset. RetinaNet-ResNet50 and RetinaNet-ResNet50-half denote the RetinaNet framework with backbones of ResNet50 and ResNet50-half, respectively. "(DKD)" means models optimized with our DKD method.

Method	Epoch	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RetinaNet-ResNet50	10	25.0	38.8	26.6	8.6	27.8	38.1
	20	29.6	44.6	31.4	11.7	33.3	43.9
	30	29.8	45.1	31.8	11.8	33.5	43.9
RetinaNet-ResNet50 (DKD)	10	26.4	40.7	28.1	10.0	29.8	39.6
	20	30.3	45.6	32.4	12.6	34.5	44.7
	30	30.8	46.0	32.9	12.6	35.1	45.4
RetinaNet-ResNet50-half	10	23.4	36.7	24.7	7.9	25.7	35.7
	20	27.4	42.4	29.5	10.0	30.8	41.7
	30	27.5	42.9	29.9	10.4	31.3	42.3
RetinaNet-ResNet50-half (DKD)	10	24.7	38.8	25.9	9.1	27.2	37.5
	20	28.0	42.8	29.8	10.7	31.3	42.4
	30	28.3	43.3	30.3	11.3	31.6	42.9

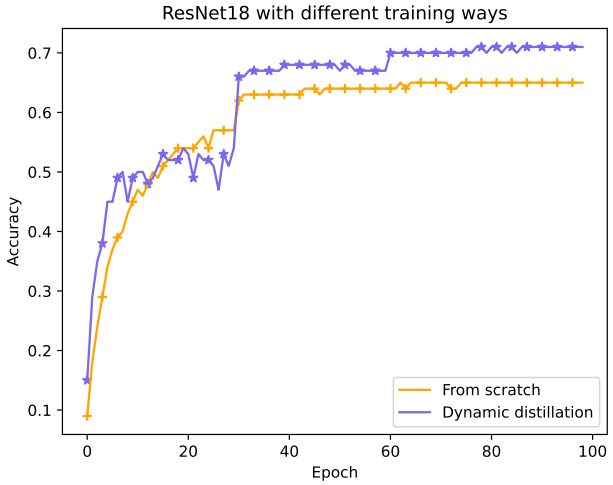

Fig. 7: Training ResNet18 with different distillation strategies. We evaluate three different training strategies, *i.e.*, static distillation, dynamic distillation, and training from scratch. We adopt the very basic training protocols without applying any targeted augmentations.

detection performance is obtained partly due to a dynamic bidirectional positive feedback flow is formed in the DKD model.

These improvements demonstrate that DKD enhances the representational capability of both the teacher and student networks through communicating their informative cues dynamically. Similar phenomenons can be also observed in Figure 7, where our DKD helps to continuously exploit and exchange meaningful information of the teacher network and the student network to boost the optimization.

We conduct a targeted ablation study on FCOS to demonstrate the detailed effects of our designs. First, we independently evaluate the effects of KLD and EMD by comparing them to the raw training strategy without applying any knowledge distillation. As shown in Table IV, compared to "ResNet50-half-FCOS", training models with KLD, EMD, and $KLD+EMD$ introduce accuracy gains by 0.8% (24.7% vs 23.9%), 1.9% (25.8% vs 23.9%), and 2.2% (26.1% vs 23.9%), respectively. It shows that both KLD and EMD are able to incorporate meaningful cues from the teacher network into the student network to compensate the learning process. However, the accuracy improvement brings by KLD is comparatively low, because KLD is responsible for distilling class probabilities, which only contains highly abstract semantic

TABLE IV: An ablation study for DKD on ResNet50-half-FCOS about object detection on MS COCO2017. "ResNet50-FCOS" and "ResNet50-half-FCOS" denote FCOS frameworks with ResNet50 and ResNet50-half backbones, respectively.

Model	MAP
ResNet50-FCOS	35.4
ResNet50-half-FCOS	23.9
ResNet50-FCOS(t)+EMD+KLD	36.0
ResNet50-half-FCOS(s)+MSE	25.0
ResNet50-half-FCOS(s)+KLD	24.7
ResNet50-half-FCOS(s)+EMD	25.8
ResNet50-half-FCOS(s)+EMD+KLD	26.1

TABLE V: Performance of DKD and SKD with the Swin Transformer model on the ILSVRC dataset. "SW-T(s)" and "SW-S(t)" represent Swin-Tiny and Swin-Small as the student network and the teacher network, respectively. "Raw Acc" represents the raw accuracy of the model without SKD and DKD.

	Model	Raw Acc	Acc	Params	FLOPs
SKD	SW-T(s)	81.0	81.4	28M	4.5G
	SW-S(t)	83.1	83.1	50M	8.7G
DKD	SW-T(s)	81.0	81.7	28M	4.6G
	SW-S(t)	83.1	83.4	51M	8.7G

TABLE VI: Performance of DKD and SKD with the Swin Transformer model on the COCO dataset. The backbone is "SW", and the model is "Mask RCNN" [57].

	Model	Raw Acc (mAP)	Acc (mAP)	Params	FLOPs
SKD	Mask RCNN + SW-T(s)	43.5	43.9	43M	279G
	Mask RCNN + SW-S(t)	46.0	46.0	54M	342G
DKD	Mask RCNN + SW-T(s)	43.5	44.1	45M	281G
	Mask RCNN + SW-S(t)	46.0	46.5	55M	344G

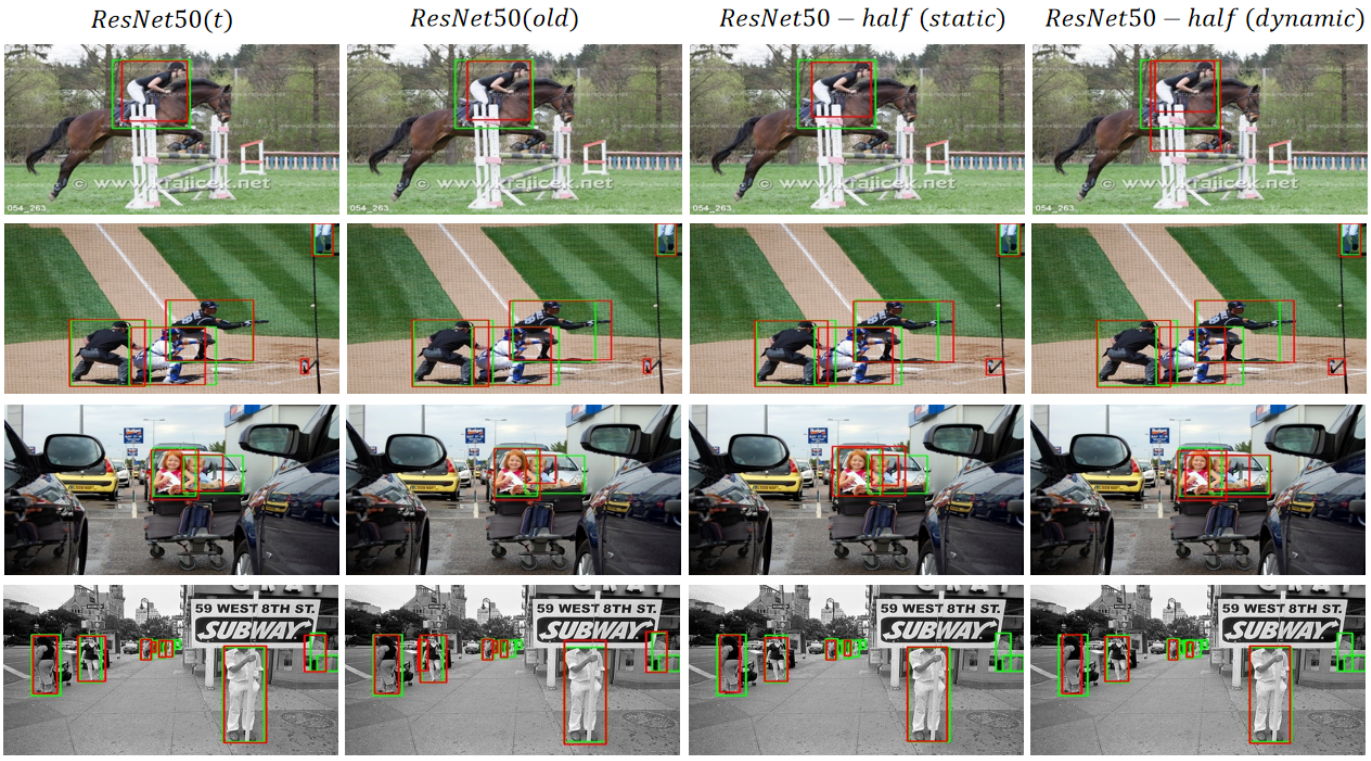

Fig. 8: Comparison of DKD and SKD with different models for object detection. We use ResNet50-FCOS as the teacher network and ResNet50-half-FCOS as the student network. "ResNet50(t)" and "ResNet50(old)" denote the effects of ResNet50-FCOS with and without DKD, respectively. Similarly, "ResNet50-half(static)" and "ResNet50-half(dynamic)" denote ResNet50-half-FCOS with SKD and DKD, respectively.

features. Additionally, we compare DKD to the existing SKD methods in Table IV, where the results display that both our explored EMD and $EMD + KLD$ outperform MSE by a clear margin (*i.e.*, 0.8% and 1.1%). Specifically, with $KLD + EMD$, "ResNet50-FCOS+KLD+EMD" achieves an accuracy gain by 0.6% (36.0% vs 35.4%) to its corresponding raw teacher network. Figure 8 shows that our DKD is able to reinforce the optimization of the teacher network and the student network jointly.

Actually, the proposed DKD is mainly used to the backbone models to improve their ability to learn useful knowledge. Specifically, we selected the state-of-the-art backbone "Swin Transformer model" [58] for testing. For the image recognition task, as shown in the Table V, on the ILSVRC dataset, for the student network, compared to "Raw Acc", our DKD obtains 0.3% (83.4% vs 83.1%) accuracy improvement while SKD with no accuracy enhancement. For the teacher network, compared to "Raw Acc", our DKD obtains 0.7% (81.7% vs 81.0%) accuracy improvement while SKD gains 0.4% (81.4% vs 81.0%) accuracy enhancement, where the improvement brings by our DKD is 75.0% (0.7% vs 0.4%) higher than SKD. For params (29M vs 28M) and FLOPS (4.6G vs 4.5G), there is no significant difference between our DKD and the traditional SKD. For the object detection task, as shown in VI on the COCO dataset, for the student network, compared to "Raw Acc", our DKD gets 0.5% (46.5% vs 46.0%) mAP accuracy improvement while SKD without any accuracy enhancement. For the teacher network, compared to "Raw Acc", our DKD gets 0.8% (44.3% vs 43.5%) mAP accuracy improvement while SKD achieves 0.4% (43.9% vs 43.5%) accuracy enhancement, where the improvement brings by our DKD is one time (0.8% vs 0.4%) higher than SKD. Again, there is not much parameters (43M vs 45M) and efficiency (279G vs 282G) difference between our DKD and the traditional SKD. In summary, compared with SKD, our DKD only changes the loss function and the training method, without changing the model structure. Consequently, the parameters and efficiency between them are insignificant.

D. Knowledge transfer on heterogeneous data

1) Experiments on DKD for video action recognition: To evaluate the performance of our proposed DKD on video action recognition, parameters β_1, β_2 and α in Eq. 8 are set to 5, 5 and 0.1, respectively. As shown in Table VII, all the evaluated knowledge distillation strategies *i.e.* "Static", "Static + Dynamic", and pure "Dynamic" can boost the accuracy of action recognition. For the RGB models on HMDB51-1 [52] and UCF101-1 [53], our "Dynamic" (DKD) strategy boosts the performance of "Static" (SKD) by 0.7% and 0.5% on accuracy, respectively. In addition, our DKD further introduces slightly gain by merging them (*i.e.* "Static + Dynamic") to train the networks. It reals that our DKD is valid to work together with SKD, and the hybrid strategy "Static + Dynamic" is a little bit better than our pure "Dynamic". Note that since UCF101 has

TABLE VII: Knowledge distillation experiments on UCF101 and HMDB51 for action recognition. HMDB51 and UCF101 are divided into three parts. HMDB51-1 and UCF101-1 denote the first part of these two datasets, respectively.

Data	Method	Flow(t)	RGB(s)
HMDB51-1	Original	75.9	73.5
	Static		78.9
	Dynamic		79.6
	Static+Dynamic	76.1	79.7
UCF101-1	Original	95.7	95.2
	Static		96.7
	Dynamic		97.2
	Static+Dynamic	95.9	97.3

TABLE VIII: Comparison of DKD with the state-of-the-art video-based action recognition methods. The accuracy on UCF101 and HMDB51 are averaged over 3 dataset splits. The results of other methods are quoted from the original papers.

Method	Pre-train	UCF101	HMDB51
C3D [42]	Sports-1M	82.3	56.8
Inception3D [59]	Kinetics	87.2	56.9
C3D+iDT [42]	Sports-1M	90.4	
I3D [43]	ImNet+Kin	94.6	74.8
ResNext101 [49]	Kinetics	94.5	70.1
S3D-G [60]	ImNet+Kin	96.8	75.9
R(2+1)D (RGB) [61]	Kinetics	96.8	74.5
STC-ResNext [59]	ImageNet	96.5	74.9
DynamoNet [62]	Kinetics	97.8	76.8
TS-Net [63]	ImageNet	88.0	59.4
TSN [64]	ImageNet	94.2	69.4
R(2+1)D (RGB+Flow) [61]	Kinetics	97.3	78.7
OFF [65]	none	96.0	74.2
TEINet-RGB [66]	ImageNet	96.7	72.1
TEA [66]	ImNet+Kin	96.9	73.3
TDN [67]	ImNet+Kin	97.4	76.3
MARS [11]	Kinetics	97.4	79.3
MARS+RGB [11]	Kinetics	97.6	79.5
RGB+Flow (ours)	Kinetics	**98.1**	**80.7**

lower difficulty than HMDB51 for action recognition, DKD brings relatively lower accuracy increases *i.e*. 2% vs 6.1%. In the action recognition task, our DKD no longer focuses on distinguishing the teacher network and the student network, instead, the two networks learn knowledge from each other to form a better, mixed network. It pays attention to fusing different source information, which refers to the motion information of the optical flow model and the appearance information of the RGB model here. In brief, these evaluations verify that the exploited DKD (1) can benefit the communication of multiple visual cues e.g. optical flow and RGB for the heterogeneous data, (2) is effective to combine with SKD and, (3) obtains higher improvement on the complicated dataset.

2) Comparison with the state-of-the-arts: We compare our DKD (RGB+Flow) model, which is simply obtained by transforming MARS [11] with using DKD to replace SKD, to the state-of-the-art video-based action recognition methods in Table VIII. In contrast to the methods that conduct knowledge distillation, our DKD works well. E.g., compared with the baseline model of us *i.e*. MARS [11], which distills knowledge by SKD, our DKD (RGB+Flow) model exceeds it by 1.4% (80.7% vs 79.3%) and 0.7% (98.1% vs 97.4%) on HMDB51 and UCF101 respectively. Compared with other methods that without conducting knowledge distillation, the exploited DKD also introduces superior performances. E.g., our DKD (RGB+Flow) model surpasses the popular R(2+1)D (RGB+Flow) model [61] by 2.0% and 0.8% on HMDB51 and UCF101, separately. Moreover, it outperforms the currently TDN model [67], which also considers the temporal motion information like our Flow model, by 4.4% and 0.7% on HMDB51 and UCF101, respectively. The experimental results validate the effectiveness of the proposed DKD for video-based human action recognition.

V. Conclusion

We propose an novel dynamic knowledge distillation (DKD) method for improving the traditional static knowledge distillation (SKD) frameworks. In contrast to existing frameworks which solely leverage the teacher network to guide the student network, our DKD enables a continuous bi-directional learning between the teacher network and the student network. We extensively testing the performance of DKD on various visual tasks, e.g. DKD is applied to the compressing model on homologous data and the knowledge transfer on heterogeneous data. As for image classification and object detection (*i.e*., evaluating on the homologous data), we show that DKD can help to exploit informative cues of features and class probabilities of the teacher network. As for video-based human action recognition (*i.e*., evaluating on the heterogeneous data), we combine the characteristics of RGB and optical flow to form a knowledge cycle in DKD, which shows further improvement over the existing methods. Significantly, we give a mathematical analysis about the convergence and efficiency of DKD. The remarkable experimental results demonstrate that the proposed DKD is effective and has a wide range of applications.

Acknowledgment

This work was supported by the Joint Fund of the Ministry of Education of China under Grant 8091B032156, the National Natural Science Foundation of China under Grant 62106177, and the fund of Tencent AI Lab RBFR2022012. The numerical calculation was supported by the super-computing system in the Super-computing Center of Wuhan University.

References

[1] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, in: International Conference on Learning Representations (ICLR), 2015.

[2] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: NIPS Deep Learning and Representation Workshop, 2015.

[3] Y. Huang, P. Shen, Y. Tai, S. Li, X. Liu, J. Li, F. Huang, R. Ji, Improving face recognition from hard samples via distribution distillation loss, in: European Conference on Computer Vision, 2020, pp. 138–154.

[4] Y. Liu, J. Yuan, Z. Tu, Motion-driven visual tempo learning for video-based action recognition, IEEE transactions on Image Processing 31 (2022) 4104–4116.

[5] Z. He, T. Dai, J. Lu, Y. Jiang, S.-T. Xia, Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution, in: IEEE International Conference on Image Processing (ICIP), 2020, pp. 518–522.

[6] A. Mishra, D. Marr, Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy, in: International Conference on Learning Representations (ICLR), 2018.

[7] M. Phuong, C. H. Lampert, Distillation-based training for multi-exit architectures, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 1355–1364.

[8] Y. Hou, Z. Ma, C. Liu, C. C. Loy, Learning lightweight lane detection cnns by self attention distillation, in: IEEE/CVF international conference on computer vision, 2019, pp. 1013–1021.

[9] Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, Self-training with noisy student improves imagenet classification, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.

[10] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: International Conference on Learning Representations (ICLR), 2017.

[11] N. Crasto, P. Weinzaepfel, K. Alahari, C. Schmid, Mars: Motion-augmented rgb stream for action recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7882–7891.

[12] Z. Huang, N. Wang, Like what you like: Knowledge distill via neuron selectivity transfer, arXiv preprint arXiv:1707.01219 (2017).

[13] Y. Zhang, T. Xiang, T. M. Hospedales, H. Lu, Deep mutual learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4320–4328.

[14] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5769–5779.

[15] J. Yim, D. Joo, J. Bae, J. Kim, A gift from knowledge distillation: Fast optimization, network minimization and transfer learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4133–4141.

[16] J. Zhu, S. Tang, D. Chen, S. Yu, Y. Liu, M. Rong, A. Yang, X. Wang, Complementary relation contrastive distillation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9260–9269.

[17] T. Wang, L. Yuan, X. Zhang, J. Feng, Distilling object detectors with fine-grained feature imitation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4933–4942.

[18] Y. Chen, Y. Xian, A. Koepke, Y. Shan, Z. Akata, Distilling audio-visual knowledge by compositional contrastive learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7016–7025.

[19] Q. Wang, L. Zhan, P. Thompson, J. Zhou, Multimodal learning with incomplete modalities by knowledge distillation, in: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1828–1838.

[20] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, Action recognition with dynamic image networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (12) (2018) 2799–2813.

[21] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, J. Yuan, Action-stage emphasized spatiotemporal vlad for video action recognition, IEEE Transactions on Image Processing 28 (6) (2019) 2799–2812.

[22] L. Tian, Z. Tu, D. Zhang, J. Liu, B. Li, J. Yuan, Unsupervised learning of optical flow with cnn-based non-local filtering, IEEE Transactions on Image Processing 29 (2020) 8429–8442.

[23] B. Heo, M. Lee, S. Yun, J. Y. Choi, Knowledge distillation with adversarial samples supporting decision boundary, in: AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3771–3778.

[24] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, Z. Dai, Variational information distillation for knowledge transfer, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9163–9171.

[25] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR), 2015.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] T.-J. Yang, Y.-H. Chen, V. Sze, Designing energy-efficient convolutional neural networks using energy-aware pruning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5687–5695.

[29] W.-C. Chen, C.-C. Chang, C.-R. Lee, Knowledge distillation with feature maps for image classification, in: Asian Conference on Computer Vision, 2018, pp. 200–215.

[30] Y. Guan, P. Zhao, B. Wang, Y. Zhang, C. Yao, K. Bian, J. Tang, Differentiable feature aggregation search for knowledge distillation, in: European Conference on Computer Vision, 2020, pp. 469–484.

[31] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[32] R. Girshick, Fast r-cnn, in: IEEE international conference on computer vision, 2015, pp. 1440–1448.

[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, Vol. 28, 2015, pp. 91–99.

[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[35] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[36] G. Chen, W. Choi, X. Yu, T. Han, M. Chandraker, Learning efficient object detection models with knowledge distillation, in: Advances in neural information processing systems, Vol. 30, 2017.

[37] J. Hoffman, S. Gupta, T. Darrell, Learning with side information through modality hallucination, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 826–834.

[38] X. Dai, Z. Jiang, Z. Wu, Y. Bao, Z. Wang, S. Liu, E. Zhou, General instance distillation for object detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7842–7851.

[39] Y. Wei, X. Pan, H. Qin, W. Ouyang, J. Yan, Quantization mimic: Towards very tiny cnn for object detection, in: European conference on computer vision (ECCV), 2018, pp. 267–283.

[40] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: IEEE conference on computer vision and pattern recognition, 2016, pp. 1933–1941.

[41] J. Stroud, D. Ross, C. Sun, J. Deng, R. Sukthankar, D3d: Distilled 3d networks for video action recognition, in: IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 625–634.

[42] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: IEEE international conference on computer vision, 2015, pp. 4489–4497.

[44] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: IEEE/CVF international conference on computer vision, 2019, pp. 6202–6211.

[45] M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks, in: International Conference on Learning Representations (ICLR), 2017.

[46] F. Hormander, N. Totaro, A. V. M. Waldschmidt, Grundlehren der mathematischen wissenschaften 332, Vol. 5, 2006.

[47] V. M. Panaretos, Y. Zemel, Statistical aspects of wasserstein distances, in: Annual review of statistics and its application, Vol. 6, 2019, pp. 405–431.

[48] F. Nielsen, On the jensen–shannon symmetrization of distances relying on abstract means, Entropy 21 (5) (2019) 485.

[49] K. Hara, H. Kataoka, Y. Satoh, Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?, in: IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6546–6555.

[50] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: IEEE conference on computer vision and pattern recognition, 2014, pp. 2147–2154.

[51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, 2014, pp. 740–755.

[52] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: International conference on computer vision, 2011, pp. 2556–2563.

[53] 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).

[54] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime tv-l 1 optical flow, in: Joint pattern recognition symposium, 2007, pp. 214–223.

[55] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 961–970.

[56] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: IEEE international conference on computer vision, 2017, pp. 2980–2988.
[57] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask r-cnn, in: IEEE international conference on computer vision, 2017, pp. 2961–2969.
[58] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.
[59] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, L. Van Gool, Spatio-temporal channel correlation networks for action classification, in: European Conference on Computer Vision (ECCV), 2018, pp. 284–299.
[60] S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification, in: European Conference on Computer Vision (ECCV), 2018, pp. 305–321.
[61] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6450–6459.
[62] A. Diba, V. Sharma, L. V. Gool, R. Stiefelhagen, Dynamonet: Dynamic action and motion network, in: IEEE International Conference on Computer Vision, 2019, pp. 6192–6201.
[63] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in neural information processing systems, 2014, pp. 568–576.
[64] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.
[65] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, W. Zhang, Optical flow guided feature: A fast and robust motion representation for video action recognition, in: IEEE conference on computer vision and pattern recognition, 2018, pp. 1390–1399.
[66] Z. Liu, D. Luo, Y. Wang, L. Wang, Y. Tai, C. Wang, J. Li, F. Huang, T. Lu, Teinet: Towards an efficient architecture for video recognition, in: AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11669–11676.
[67] L. Wang, Z. Tong, B. Ji, G. Wu, Tdn: Temporal difference networks for efficient action recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1895–1904.

Zhigang Tu started his Master Degree at Wuhan University, China, 2008. In 2015, he received the Ph.D. degree from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. Then from 2016 to 2018, he was a research fellow at Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Special for motion estimation (optical flow), action recognition and localization, human/hand pose estimation, and anomaly event detection.

He has co-/authored more than 60 articles on international SCI-indexed journals and conferences. He is an Associate Editor of the SCI-indexed journal *The Visual Computer* (IF=2.835) and a Guest Editor of *Journal of Visual Communications and Image Representation* (IF=2.887), the Area Chair of AAAI2023 and VCIP2022. He is the first organizer of the ACCV2020 Workshop on MMHAU (Japan). He received the "Best Student Paper" Award in the 4^{th} Asian Conference on Artificial Intelligence Technology.

Xiangjian Liu received his Master Degree in image processing at Wuhan University, China, 2019. He received the B. Eng. degree from the school of geographic information science from Jilin University in 2019. His research interests mainly include computer vision and machine learning.

Xuan Xiao , a medical doctor, professor, master supervisor, and vice president of Renmin Hospital, Wuhan University, China. Professor Xiao received the Ph.D in clinical medicine from Wuhan University in 2009. She is mainly engaged in treatment and management of fundus diseases, with a focus on the prevention and treatment of central retinal artery occlusion (CRAO) and its derived pan-vascularized ophthalmic cardiovascular and cerebrovascular events (OHBI).