FLRKD: Relational Knowledge Distillation Based on Channel-wise Feature Quality Assessment

Zeyu An[1]
ZDAzy0219@163.com
Changjian Deng[1]
cjdeng@std.uestc.edu.cn
Wanli Dang[1,2]
dangwanli@caacsri.com
Zhicheng Dong[3]
dongzc666@163.com
Qian Luo[2]
caacsri_luoqian@163.com
Jian Cheng†[1]
chengjian@uestc.edu.cn

[1] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2] The Second Research Institute of Civil Aviation Administration of China, Chengdu, China

[3] School of Information Science and Technology, Tibet University, Lhasa, China

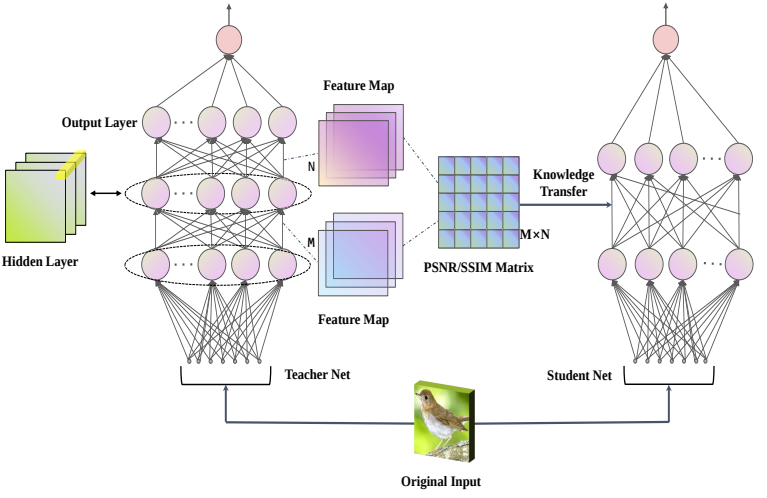
Abstract

With the increasing computational power of computing devices, the pre-training of large deep-learning models has become prevalent. However, deploying such models on edge devices with limited memory and computing power remains a significant challenge. To address this issue, this study proposes a novel knowledge distillation approach called Feature-level Relationship-based Knowledge Distillation (FLRKD). The proposed approach employs image quality similarity assessment to distill knowledge from a pre-trained model into smaller models that are suitable for deployment on edge devices. FLRKD utilizes peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) between feature maps of different hidden layers as relational knowledge to enhance the classification accuracy of student models. Moreover, the proposed approach includes an effective loss function that accelerates the convergence of the knowledge distillation algorithm. Additionally, a regressor is introduced to address the issue of inconsistent feature map spatial size between teacher and student models in heterogeneous scenarios. Comparative and ablation experiments demonstrate the superiority of FLRKD over mainstream knowledge distillation methods in terms of higher classification accuracy (up to 4%) and faster convergence rates. Notably, the proposed approach achieves significant improvement in classification accuracy (up to 3%) even in heterogeneous scenarios compared to existing state-of-the-art methods.

1 Introduction

This paper introduces the concept of knowledge distillation (KD)[10], which transfers knowledge from a large model to a smaller one to reduce the computational burden. KD can be divided into three types: response-based, feature-based, and relationship-based. Relationship-

Figure 1: Conceptual diagram of the FLRKD method. The PSNR matrix and the SSIM matrix represent the knowledge extracted from the pre-trained deep neural network, which are obtained by calculating the peak sign-to-noise ratio and the SSIM between the feature maps of two different hidden layers. The flow between two layers can be represented by the PSNR matrix and the SSIM matrix.



based KD examines the relationships between different layers in the network and the input/output data. Existing methods such as the FSP and graph-based methods have limitations. This paper proposes a new method that incorporates the PSNR[30] and the SSIM[29] as a measure of similarity between feature maps to generate a PSNR matrix and SSIM matrix to represent the flow in the network inference process. The proposed approach aims to introduce image quality similarity as a metric for similarity measurement in relationship-based KD. This method can capture more complex feature relationships and may improve knowledge transfer performance[9, 10, 11, 16, 17, 18, 19, 26, 31, 32, 34]. Considering the significance of spatial structure information in the neural network training process[1], it has been observed that existing relationship-based knowledge distillation algorithms, such as FSP[32], solely focus on the correlation among points in different feature maps. Unfortunately, they neglect the correlation of spatial structure information across different feature maps. This limitation serves as the driving force behind the algorithm proposed in this research paper. In terms of similarity metrics, it has been discovered that the combination of PSNR and SSIM not only takes into account the correlation between points but also considers the variability in spatial structure information within feature maps[23]. Figure 1 shows a conceptual diagram of our proposed refined knowledge transfer approach.

The main contributions of this paper are as follows:

1. We propose a new type of knowledge that is characterized by the PSNR matrix and SSIM matrix computed between different channels in different hidden layers, which has a better effect on guiding student model training than the knowledge used in the current mainstream relationship-based knowledge distillation methods.

2. Employing the proposed refinement approach to obtain initial weights can effectively

†Corresponding author
© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

enhance the performance of compact neural networks, while also yielding a quicker convergence rate.

3. Unlike other relationship-based knowledge distillation methods, our approach can also be applied when the teacher and student models are heterogeneous. Even if the structure of the student model differs from that of the teacher model, our proposed method can significantly improve the performance of the student model.

2 Related Work

This paper focuses on knowledge distillation methods, which aim to reduce the number of deep network parameters while maintaining the original performance level. Knowledge distillation is achieved by transferring knowledge from pre-trained large models to smaller networks, and recently, relationship-based knowledge distillation methods have been introduced, which use different techniques to model the relationships between data samples, such as cross-sample similarity knowledge, and distillation loss of distance and angle. In recent years, there have been emerging methodologies in the field that integrate quantization techniques with knowledge distillation. Additionally, there are methods that focus on aligning the encoder characteristics of the teacher-student network to facilitate effective knowledge distillation. By leveraging the similarities in encoding mechanisms, these techniques enable efficient knowledge transfer. Furthermore, certain strategies aim to improve knowledge distillation by discerning and addressing the discrepancies between deep and shallow task differences within the network architecture. Moreover, there is ongoing research dedicated to enhancing the performance of knowledge distillation by specifically targeting the improvement of the logit, the pre-softmax output of the model[2, 4, 5, 6, 8, 10, 11, 12, 15, 16, 19, 20, 22, 25, 26, 28, 32, 34, 35].

3 Method

The main basis of our proposed method is how to better define the knowledge information of the teacher model and transfer the extracted knowledge to the student model. In this case, the student model not only has excellent performance of the teacher model but also has fewer parameters. This section is divided into four parts to illustrate our main concepts. Section 3.1 introduces the new types of flow and knowledge used in this work. Section 3.2 provides a more objective description of the knowledge used in this paper through mathematical expressions. Based on the well-designed and refined knowledge, we define the loss function in Section 3.3. Finally, Section 3.4 introduces the whole process of classroom student network in this work.

3.1 Feature-level Relationship-based Knowledge

Response-based knowledge focuses on the output results of the softmax layer, which is result-oriented. However, this approach overlooks the knowledge contained in the hidden layer of the neural network. On the other hand, feature-based knowledge distillation methods imitate the intermediate results of the feature layer of the teacher network but fail to consider the spatial relationship between shallow and deep layers. To address these issues, we propose a knowledge distillation method based on the flow of problem-solving processes

represented by PSNR and SSIM matrices between feature maps from different layers. Our method avoids the problem of selecting effective feature maps and accurately represents the hierarchical relationships in the model. We use PSNR and SSIM to measure the feature map similarity, which is equivalent to measuring the similarity of characteristic distributions in an RGB image. Our experimental results confirm the validity and feasibility of this approach.

3.2 Mathematical Expression of the Distilled Knowledge

The relationship between the two different hidden layer feature maps can be used to define the flow of the solution process. In this case, the relationship between both of them is described by the PSNR and SSIM. We construct the PSNR matrix and the SSIM matrix to represent the flow of the solution process. The PSNR matrix $P \in R^{m\times n}$ and the SSIM matrix $S \in R^{m\times n}$ are generated by the feature of two layers.

3.2.1 PSNR and SSIM

The PSNR[30] is an extension of the Mean Square Error (MSE) in the image and is one of the indicators used to measure image quality. More precisely, this parameter is the statistical analysis of the grey value of the image pixels. For two different images $\mathcal{I}$ and $\mathcal{K}$, both of which have a size of $m \times n$, their mean square error is expressed by the following formula:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|\mathcal{I}(i,j) - \mathcal{K}(i,j)\| \tag{1}$$

PSNR is defined as:

$$PSNR = 10\cdot\log_{10}\left(\frac{MAX}{MSE}\right) = 20\cdot\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{2}$$

Where MSE is the mean square error of the current image $\mathcal{I}$ and $\mathcal{K}$, and MAX is the maximum image pixel value. If each pixel is represented by an 8-bit integer, its value is 255. The higher the PSNR[30] value, the lower the distortion.

PSNR[30] is a commonly used objective index for evaluating image quality, but it only considers the error between corresponding pixels, which neglects the spatial information of the image. Therefore, this study introduces SSIM as a full-reference image quality metric that accounts for image brightness, contrast, and structure, and emphasizes the interdependence between adjacent pixels and object structure information.

Similarly, for the given two images $\mathcal{I}$ and $\mathcal{K}$, their SSIM is defined as the following formula:

$$SSIM = [l(\mathcal{I},\mathcal{K})]^{\alpha}[c(\mathcal{I},\mathcal{K})]^{\beta}[s(\mathcal{I},\mathcal{K})]^{\gamma} \tag{3}$$

$$l(\mathcal{I},\mathcal{K}) = \frac{2\mu_{\mathcal{I}}\mu_{\mathcal{K}} + C_1}{\mu_{\mathcal{I}} + \mu_{\mathcal{K}} + C_1} \tag{4}$$

$$c(\mathcal{I},\mathcal{K}) = \frac{2\sigma_{\mathcal{I}}\sigma_{\mathcal{K}} + C}{\sigma_{\mathcal{I}} + \sigma_{\mathcal{K}} + C} \tag{5}$$

$$s(\mathcal{I},\mathcal{K}) = \frac{\sigma_{\mathcal{I}\mathcal{K}} + C_3}{\sigma_{\mathcal{I}}\sigma_{\mathcal{K}} + C_3} \tag{6}$$

Where $l(\mathcal{I},\mathcal{K})$ represents the brightness of $\mathcal{I}$ and $\mathcal{K}$, $c(\mathcal{I},\mathcal{K})$ represents the contrast of $\mathcal{I}$ and $\mathcal{K}$, $s(\mathcal{I},\mathcal{K})$ represents the structure of $\mathcal{I}$ and $\mathcal{K}$, α ,β ,γ are the parameter adjustment of SSIM, which is used to adjust the importance between the three modules, and all are non-negative values. $\mu_{\mathcal{I}}$ and $\mu_{\mathcal{K}}$ respectively represent the mean values of images $\mathcal{I}$ and $\mathcal{K}$,$\sigma_{\mathcal{I}}$ and $\sigma_{\mathcal{K}}$ respectively represent the variances of images $\mathcal{I}$ and $\mathcal{K}$,$\sigma_{\mathcal{I}\mathcal{K}}$ represents the covariances of images $\mathcal{I}$ and $\mathcal{K}$, and C_1,C,C_3 are constants. The value of SSIM is in the range of 0 to 1. The higher the value, the better the quality.

The constant C_1 is to avoid the instability of the system when $\mu_{\mathcal{I}}+\mu_{\mathcal{K}}$ tends to 0. In particular, we set C_1 to (K_1L), where L is the number of image gray levels, and for 8-bit grayscale images, L equals 255 and $K_1 \ll 1$. Similarly, we set C to (K_1L) and $K \ll 1$. To simplify the calculation, we set all three hyperparameters of α ,β ,γ to 1, and let $C_3 = \frac{C}{2}$. Then, Then, we can get the simplified form of SSIM.

$$SSIM = \frac{(2\mu_{\mathcal{I}}\mu_{\mathcal{K}}+C_1)(2\sigma_{\mathcal{I}\mathcal{K}}+C)}{(\mu_{\mathcal{I}}+\mu_{\mathcal{K}}+C_1)(\sigma_{\mathcal{I}}+\sigma_{\mathcal{K}}+C)} \tag{7}$$

3.2.2 PSNR Matrix and SSIM Matrix

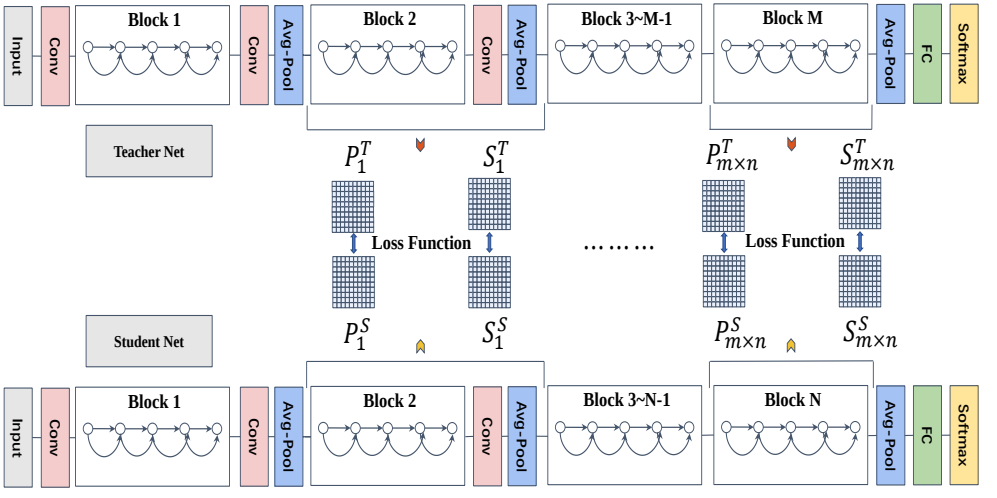
First, we assume that there are M blocks in the teacher network and N blocks in the student network. Arbitrarily select mutually exclusive blocks from the student network, and accordingly, we also extract η ($\eta \leq N \leq M$) mutually exclusive blocks from the teacher network, and the mapping relationship between the teacher network and the student network blocks is established in accordance with $\sigma \cdot \lfloor \frac{M}{N} \rfloor$ ($\sigma \in R^+, \sigma = 1,2,\cdots,\eta$). Then the input feature maps $F^T_{\sigma-1} \in R^{h\times w\times a}$,$F^S_{\sigma-1} \in R^{h\times w\times b}$ and the output feature maps $F^T_{\sigma} \in R^{h\times w\times c}$,$F^S_{\sigma} \in R^{h\times w\times d}$ (where h, w, and a,b,c,d represent the height, width, and number of channels of the feature map)of the blocks selected from the teacher-student network are extracted and the PSNR/SSIM matrix is computed as follows:

$$P_{a/b,c/d}(x;W) = 10\cdot\log_{10}\left(\frac{MAX}{\frac{1}{hw}\sum_{s=0}^{h-1}\sum_{t=0}^{w-1}\left\|F^{T/S}_{s,t,i}(x;W)-F^{T/S}_{s,t,j}(x;W)\right\|}\right) \tag{8}$$

$$\begin{aligned} S_{a/b,c/d}(x;W) = &\left[l\left(F^{T/S}_{s,t,i}(x;W),F^{T/S}_{s,t,j}(x;W)\right)\right]^{\alpha} \\ &\times\left[c\left(F^{T/S}_{s,t,i}(x;W),F^{T/S}_{s,t,j}(x;W)\right)\right]^{\beta} \\ &\times\left[s\left(F^{T/S}_{s,t,i}(x;W),F^{T/S}_{s,t,j}(x;W)\right)\right]^{\gamma} \end{aligned} \tag{9}$$

Where x and W represent input image and deep neural network weights, where a,c,T are correspondences. b,d,S are correspondences. Where i and j are the index values of the different channels that correspond to the feature maps of the two hidden layers. s and t are the line number and column number used to calibrate the specific position of the feature. In particular, the selected location of the block of the student network is arbitrary, it is only necessary to ensure that the block selection in the teacher network satisfies the above mapping relationship. we select several different locations for the generation of the PSNR matrix and the SSIM matrix. The specific process of the operation is shown in figure 2.

Figure 2: Framework of our FLRKD algorithm when the ResNet structure is used in the teacher and student networks. The teacher network and the student network can adopt a variety of network architectures, provided that the size of the feature maps of the student network and the teacher network are the same. (To ensure the same size of the two feature maps, a regression is introduced in this paper). There are two stages to our proposed method. In stage 1, the student network is trained to minimize the distance between the PSNR and SSIM matrices of the student and teacher networks. Then, the pre-trained weights of the student deep neural network are used for the initial weight in stage 2. Stage 2 is the normal training procedure[32].



3.3 Loss for the FLRKD

The above indicators all aim at transferring knowledge more effectively from the large pre-trained teacher model to the small student model. To achieve this, we take the PSNR matrix and the SSIM matrix proposed in section 3.2.2 as knowledge. This is the aforementioned flow in the inference process of the network. It has been demonstrated the teacher network has its corresponding PSNR matrix and SSIM matrix, and the student network also has its corresponding PSNR matrix and SSIM matrix. As the network structure between the teacher models and the student models may differ, the feature maps' size may also be different, which will lead to different spatial sizes of the PSNR matrix and the SSIM matrix generated by the feature map. Therefore, we added a regressor after the hidden layer of the student network for feature dimension enhancement to match the feature output of the hidden layer of the teacher network.

The idea of the $\mathcal{L}_{HT}$ function comes from the work of Romero et al.[22]. The calculation of $\mathcal{L}_{HT}$ is shown as follow,

$$\mathcal{L}_{HT}\left(W_G, W_r\right) = \frac{1}{2}\left\|u_i\left(x; W_H\right) - r\left(v_j\left(x; W_G\right); W_r\right)\right\| \tag{10}$$

where u_i and v_j are the teacher/student deep nested functions up to their respective teacher/student hidden layers with parameters W_H and W_G, r is the regressor function on top of the guided layer with parameters W_r.

$\mathcal{L}_{KD}$ is determined according to the knowledge distillation method used in the task,

$$\mathcal{L}_{KD}(W_S) = \mathcal{H}(y_{true}, P_S) + \rho \mathcal{H}(P_T^{\tau}, P_S^{\tau}) \tag{11}$$

where $P_T = softmax(\alpha_T)$ where α_T is the vector of teacher pre-softmax activation. Student network output probability $P_S = softmax(\alpha_S)$, where α_S is the student's pre-softmax output. The student network will be trained such that its output P_S is similar to the teacher's output P_T, as well as to the true labels y_{true}. Since P_T might be very close to the one hot code representation of the sample's true label, a relaxation $\tau > 1$ is introduced to soften the signal arising from the output of the teacher net[11, 22].

Suppose that the PSNR matrix and the SSIM matrix of the teacher network are $P_i^T, i = 1, ..., \eta$, $S_i^T, i = 1, ..., \eta$, the corresponding PSNR matrix and the SSIM matrix of the student network are $P_i^S, i = 1, ..., \eta$, $S_i^S, i = 1, ..., \eta$. Through the function of the above regressors, the hidden layer feature maps of the teacher network and the student network have the same spatial size, so the PSNR matrix and the SSIM matrix of the teacher network and the student network have the same spatial size. In this work, $\mathcal{L}_2$ loss is used as the loss function of these two sets of parameters, which can be expressed as follows,

$$\mathcal{L}_{PSNR}(W_T, W_S) = \frac{1}{N} \sum_{x} \sum_{i=1}^{\eta} \lambda_i \times \left\| P_i^T(x; W_T) - P_i^S(x; W_S) \right\|_2 \tag{12}$$

$$\mathcal{L}_{SSIM}(W_T, W_S) = \frac{1}{N} \sum_{x} \sum_{i=1}^{\eta} \lambda_i \times \left\| S_i^T(x; W_T) - S_i^S(x; W_S) \right\|_2 \tag{13}$$

where λ_i and N are the weight and data points of each lost item. In this work, there is no distinction between the importance of different point loss items, that is, all loss items are equally important. Therefore, we use the same λ_i for all experiments.

4 Experiments

4.1 Same-Model Transfer

The dataset used in this study is CIFAR-100[14], which consists of 50,000 training images with 500 images per class and 10,000 test images. We conduct experiments on various student-teacher combinations with different capacities[14], such as ResNet[10], Wide ResNet (WRN)[33], and VGG[24]. For comparison, we use the following models: KD[11], FitNets[22], SP[28], RKD[19], PKT[21], FT[13], FSP[32], QuEST[12],SimKD[5], TDD[25], DKD[35].

To evaluate the impact of the number of negatives on performance, we vary the value of N and experiment with values of 64, 256, 1024, 4096, and 16384. Our results indicate that larger values of N lead to improved performance, and we, therefore, adopt a value of 16384 as the default for reporting accuracy. The default value for the learning rate is 0.05, and we set the learning rate decay rate to 0.1. Temperature values between 0.05 and 0.2 are found to work well on CIFAR-100[14], and we use an equal weight of 1.0 for the cross-entropy loss between logit and ground truth, as well as for the KD loss. In addition, we set the weight of the PSNR and SSIM losses for the distillation loss to 0.8.

To carry out the experiments, we employ multiple GPUs, including four 1080Ti devices.

Result Table 1 presents average accuracy results for knowledge distillation algorithms on the CIFAR-100 dataset[14] using identical architectures for both teacher and student networks. In seven experiments, FLRKD demonstrated superior performance compared to most state-of-the-art algorithms in two of them. When compared to all other relationship-based knowledge distillation algorithms, all seven experiments achieved the optimal value. Moreover, in comparison to FSP[32], the accuracy witnessed a nearly 3% increase. Combining PSNR and SSIM matrices enhanced the performance of knowledge distillation. As can be seen from the penultimate third row of Table 1, FLRKD combined with KD demonstrated superior performance compared to most state-of-the-art algorithms in three of them, while the other four experiments did not reach optimal but nearly sub-optimal values. Although FLRKD is not optimal for all knowledge distillation algorithms in a few cases, the starting point of this work is to propose a more efficient relationship-based knowledge distillation algorithm and to provide a feasible new idea for relationship-based knowledge distillation algorithms[27]. These findings suggest that FLRKD is an effective knowledge distillation approach that can improve performance when combined with other distillation algorithms[7]. The last two rows in Table 1 show the results of the ablation experiment. The results show that the performance of the FLRKD algorithm decreases by about 2% when the PSNR metric is removed, and by about 3% when the SSIM metric is removed. This shows that both PSNR and SSIM have important roles in the FLRKD algorithm and contribute significantly to both the improvement of reconstruction quality and the optimization of algorithm performance.

Table 1: Test accuracy of student networks on CIFAR-100[14] of several distillation methods (ours is FLRKD). We note that the FLRKD algorithm has the best performance among the relationship-based knowledge distillation algorithms, and we have marked relationship-based knowledge distillation algorithms in purple font in the table. The data in bold in the table correspond to the highest classification accuracy and the data in green font in the table correspond to the second-best classification accuracy obtained by using different knowledge distillation algorithms with the same teacher-student network frameworks and the same experimental conditions.

Teacher Student	wrn-40-2 wrn-16-2	wrn-40-2 wrn-40-1	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32×4 resnet8×4	vgg13 vgg8
Teacher Model	75.61	75.61	72.34	74.31	74.31	79.42	74.64
Student Model	73.26	71.98	69.06	69.06	71.14	72.50	70.36
KD[11]	74.92	73.54	70.66	70.67	73.08	73.33	72.98
FitNets[22]	73.55	72.31	69.21	69.00	71.10	73.49	71.07
SP[28]	73.38	72.40	69.63	70.12	72.70	72.93	72.71
RKD[19]	73.32	72.18	69.48	69.32	71.79	71.88	71.53
PKT[21]	74.64	73.49	70.33	70.31	72.54	73.69	72.89
FT[13]	73.12	71.60	69.76	70.26	72.37	72.81	70.58
FSP[32]	72.78	72.51	69.91	70.07	71.83	72.58	70.28
QuEST[2]	74.12	73.51	69.86	69.99	72.74	73.31	72.92
SimKD[6]	74.76	73.57	70.25	70.63	73.06	**74.52**	73.14
TDD[25]	74.73	73.44	70.00	70.52	72.98	73.10	72.94
DKD[35]	**75.02**	73.89	**70.74**	70.81	73.15	74.48	**73.96**
FLRKD(ours)	74.14	73.62	70.35	70.97	73.22	73.56	73.17
FLRKD+KD	74.96	**74.06**	70.68	**71.15**	**73.24**	73.61	73.83
w/o SSIM	73.44	72.97	69.76	70.56	73.02	73.39	71.60
w/o PSNR	73.92	73.09	69.41	70.83	73.11	73.40	72.06

4.2 Cross-Model Transfer

Result Table 2 presents the average accuracy of different knowledge distillation methods on CIFAR-100[14] dataset with varying teacher-student architectures. Based on the experi-

Table 2: Test accuracy of student networks on CIFAR100 of a number of distillation methods (ours is FLRKD) for transfer across different teacher and student architectures. Importantly, some methods that require very similar student and teacher architectures perform quite poorly. E.g. FSP [33] cannot even be applied. FLRKD can also be adapted to the Teacher-Student model using different architectures by introducing regressors[27]. The meanings of different fonts in Table 2 are consistent with those in Table 1.

Teacher	vgg13	ResNet50	ResNet50	resnet32×4	resnet32×4	wrn-40-2
Student	MobileNetV2	MobileNetV2	vgg8	ShuffleNetV1	ShuffleNetV2	ShuffleNetV1
Teacher Model	74.64	79.34	79.34	79.42	79.42	75.61
Student Model	64.60	64.60	70.36	70.50	71.82	70.50
KD[11]	67.37	67.35	73.81	74.07	74.45	74.83
FitNets[22]	64.14	63.16	70.69	73.59	73.54	73.73
SP[28]	66.30	68.08	73.34	73.48	74.56	74.52
RKD[19]	64.52	64.43	71.50	72.28	73.21	72.21
PKT[21]	67.13	66.52	73.01	74.10	74.69	73.89
FT[13]	61.78	60.99	70.29	71.75	72.50	72.03
QuEST[12]	67.92	67.76	73.80	74.58	74.65	75.45
SimKD[5]	68.45	68.33	74.33	75.12	75.91	76.36
TDD[25]	68.61	68.09	74.28	74.96	74.88	75.47
DKD[35]	69.43	68.54	74.32	**75.33**	75.83	**76.59**
FLRKD(ours)	68.59	68.11	74.42	75.17	75.77	76.02
FLRKD+KD	**69.81**	**69.24**	**75.28**	75.32	**76.00**	**76.59**
w/o SSIM	68.01	67.52	73.60	74.48	74.75	75.14
w/o PSNR	68.13	67.99	74.03	75.03	75.18	75.74

mental findings presented in Table 2, it is observed that the FLRKD combined with the KD algorithm achieves generally superior accuracy than other knowledge distillation methods, even in scenarios where the teacher-student model has distinct architectures. Specifically, in six experiments with independent variations in teacher and student network architectures, the FLRKD combined with the KD algorithm obtains the highest classification accuracy in five experimental groups. Notably, when wrn-40-2 and ShuffleNetV1 are used as teacher and student architectures respectively, FLRKD yields an accuracy improvement of nearly 2.5% over RKD[32]. Distilling intermediate representations tends to result in lower accuracy when switching from the same to different architectures, potentially due to distinct solution paths. PKT[21], SP[28] and SimKD[5] perform better by distilling from the last few layers, while FitNets[22] even perform worse than the vanilla student. The last two rows show that the performance of the FLRKD algorithm decreases by about 1% when the PSNR metric is removed, and by about 2% when the SSIM metric is removed. This shows that both PSNR and SSIM have important roles in the FLRKD algorithm and contribute significantly to both the improvement of reconstruction quality and the optimization of algorithm performance.

4.3 Fast Convergence

Pre-training deep neural networks has become a trend in recent years, but as network size increases, training time also increases[32]. Despite this, researchers continue to develop better models due to the excellent performance of deep neural networks in various fields. Therefore, there is a growing demand for fast and lightweight technology. In our proposed technique, we used one teacher network to generate several student networks, aiming to achieve similar performance with less training time than the normal procedure[5, 32]. The

Table 3: Comparison of training time and the number of model parameters after distillation of KD and some mainstream relationship-based knowledge algorithms. The original model we use here is ResNet50.

Algorithms	Original Model	Distillation Model	Distillation time
FSP[32]	98.0 MB	10.9 MB	104 min
RKD[19]	98.0 MB	9.9 MB	71 min
TDD[25]	98.0 MB	13.0 MB	105 min
DKD[35]	98.0 MB	9.3 MB	97 min
FLRKD(ours)	98.0 MB	**11.3 MB**	**82 min**

experimental results in Table 3 show that the FLRKD algorithm achieves accuracy comparable to state-of-the-art knowledge distillation methods but with a shorter distillation time. In comparison to TDD[25], and DKD[35], FLRKD exhibits superior convergence speed and achieves desirable accuracy.

5 Conclusion

In this work, we propose a new method of relationship-based knowledge distillation, based on new forms of knowledge representation. This knowledge is expressed by the PSNR matrix and the SSIM matrix, which represent information about the network inference process, i.e. the flow defined in this work. We verify the superiority, compatibility, and feasibility of FLRKD through three different sets of experiments. The experimental results show that FLRKD combined with the KD algorithm is superior to the most advanced relationship-based knowledge distillation method. In addition, there are still some problems to be solved in this work, such as how to select a more reasonable location for the hidden layer of the feature map extraction. If we put the perspective on the whole lightweight work, how to combine FLRKD with pruning or other lightweight methods, etc., all these are the problems we need to further think about and solve in the follow-up work.

6 Acknowledgments

This research was supported by the NNSFC&CAAC(No.U2133211 and No.U2233209).

References

[1] Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[2] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi. Label refinery: Improving imagenet classification through label progression, 2018.

[3] Chengpeng Chen, Zichao Guo, Haien Zeng, Pengfei Xiong, and Jian Dong. Repghost: A hardware-efficient ghost module via re-parameterization. *arXiv preprint arXiv:2211.06088*, 2022.

[4] D. Chen, J. P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen. Cross-layer distillation with semantic calibration. In *National Conference on Artificial Intelligence*, 2021.

[5] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11933–11942, 2022.

[6] Y. Chen and N. Wang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. 2017.

[7] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pages 1–15. Springer, 2000.

[8] T. Furlanello, Zachary C Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born again neural networks, 2018.

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38–39, 2015.

[12] Himalaya Jain, Spyros Gidaris, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Quest: Quantized embedding space for transferring knowledge. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 173–189. Springer, 2020.

[13] J. Kim, S. Park, and N. Kwak. Paraphrasing complex network: Network compression via factor transfer, 2018.

[14] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25(2), 2012.

[16] S. Lee and B. C. Song. Graph-based knowledge distillation by multi-head attention network, 2019.

[17] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–350, 2018.

[18] H. Noh, P. H. Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction, 2016.

[19] W. Park, D. Kim, Y. Lu, and M. Cho. Relational knowledge distillation. 2019.

[20] N. Passalis and A. Tefas. Probabilistic knowledge transfer for deep representation learning. 2018.

[21] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[22] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *Computer ence*, 2015.

[23] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. , 7(3):11, 2019.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[25] Jie Song, Haofei Zhang, Xinchao Wang, Mengqi Xue, Ying Chen, Li Sun, Dacheng Tao, and Mingli Song. Tree-like decision distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13488–13497, 2021.

[26] C. Szegedy, L. Wei, Y. Jia, P. Sermanet, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation, 2019.

[28] F. Tung and G. Mori. Similarity-preserving knowledge distillation. 2019.

[29] Z. Wang. Image quality assessment : From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[30] S. Wolf. Reference algorithm for computing peak signal to noise ratio (psnr) of a video sequence with a constant delay. *ITS*, 2009.

[31] Q. Wu, D. Teney, P. Wang, C. Shen, A. R. Dick, and Avd Hengel. Visual question answering. *Computer Vision and Image Understanding*, 2017.

[32] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] S. Zagoruyko and N. Komodakis. Wide residual networks. 2016.

[34] C. Zhang and Y. Peng. Better and faster: Knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification, 2018.

[35] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.