Optimized Computation Combining Classification and Detection Networks with Distillation

1st Meng Xu
School of Electrical Engineering and Computer Science
Queen Mary University of London
London, U.K.
meng.xu@qmul.ac.uk

2nd Yang Gu
Momenta Suzhou
Momenta
Suzhou, China
guyang@momenta.ai

3rd Stefan Poslad
School of Electrical Engineering and Computer Science
Queen Mary University of London
London, U.K.
stefan.poslad@qmul.ac.uk

4th Shiqing Xue
School of Computer Science and Engineering
Beihang University
Beijing, China
sqxue@foxmail.com

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9534331

***Abstract*—A Convolutional neural network (CNN) has emerged as a widely used approach to computer vision tasks, including object classification and detection tasks. The high requirement for the model to be more computationally efficient on lower information and communication technology (ICT) resource, e.g., mobile terminals can benefit from model distillation. However, most existing distillation methods suffer from a significant accuracy reduction, which requires a large number of pre-training models or doesn't make good use of the more of the network information, e.g., in the middle layers, during the distillation. In this paper, we study how knowledge about traffic signs recognition could be transferred to smaller models by distillation while cutting channels. We present an optimized object detection network, which uses a Region Proposal Network (RPN) weighted loss and hard-soft distribution-wise distillation loss for structural differences between teacher and student networks. We validate the network on multiple real-world datasets, the experiments demonstrate that the classification accuracy can be improved by 9% with about 16 times parameter reduction while the detection network performance could be increased by 10.6% using an optimized object detection network.**

***Index Terms*—object detection, knowledge distillation, teacher network, student network, pruning**

I. Introduction

Classification and object detection tasks are fundamental in the computer vision research area, one particularly promising research direction is model compression for these tasks is the problem of reducing the model size to show better or similar evaluation results with fewer parameter amounts compared to the original models, which could reduce models' storage and calculation pressure and is more easily to deploy.

Recent advances in computer vision have largely been driven by deep neural networks (DNN) [1] to improve the accuracy of object detection [2]and image classification task [3] has been improved greatly by using DNN, which could replace the use of traditional hand-crafted feature selection methods [4].

TABLE I
Common Classic Convolutional Neural Network Models

Model name	**Model size(MB)**	**Calculations (million)**	**No. of parameters(million)**
AlexNet [5]	>200	720	60
VGG16 [6]	>500	15300	138
GoogleNet [7]	50	1550	6.8
Inception-v3 [8]	90-100	5000	23.2

Traffic sign recognition is an important part of road transport applications to increase the safety of semi-autonomous and autonomous vehicle travel, yet they are not so easy to visually recognize and can consume a huge memory during computation and storage. Along with the requirement of high performance for object detection and classification tasks, low latency and fast processing speed are needed for further applications, such as mobile apps and autonomous cars. Although introducing more layers and more parameters often improves the accuracy of a model, the use of large-scale data and more complex DNN layered models increases the computation cost and memory use, which makes big models computationally too expensive to be deployed on lower resource devices such as mobile devices and embedded devices. In addition, the transportation and calculation speed of models are affected due to redundant parameters. In fact, some parameters have little influence in the calculation process, but may cause problems such as gradient dispersion, overfitting, and accuracy degradation. Table I. summarizes the model size, calculation amount, and number of parameters used with some classic DNN models.

With large CNN, models could obtain effective information, which is important for smart traffic and is the basis for high-level tasks. However, the task of recognizing traffic signs to obtain current road conditions in autonomous driving mode

This work is done during Meng Xu's internship at Didi Chuxing.

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

has high real-time requirements, which needs small size models for computation and storage in mobile devices, such as phones and embedded devices. Compressing models could use fewer parameters and yet achieve a high accuracy in CNNs, which can effectively deploy models in low ICT [9] resource devices.

Model compression techniques have emerged to address such issues, e.g., parameter pruning and sharing, low-rank factorization [10] and knowledge distillation [11]. Knowledge distillation is an effective technique to teach a small network (student) using a larger neural network (teacher). The small network is trained to mimic the large network's behaviour by adding supervision functions. However, most existing compression methods suffer from a significant accuracy reduction, which requires a large number of pre-training models or do not make good use of the loss in the middle network supervision. So, we propose an optimized object detection network, which uses RPN weighted loss and hard-soft distribution-wise distillation loss for structural differences between teacher and student network. We validate the network on multiple real datasets, which shows the proposed method could overpass the original methods in some performances. The experiments systematically compared how different distillation parameters and strategy applications could affect the distillation performance. Parts of the core code could be found in https://github.com/MengXu-u/Knowledge-Distillation.

In the paper, we make the following contributions:

- We critically review common methods for model compression, and make a detailed classification with characteristic analysis, and to find the inherent connection between them to apply pruning as a part of the distillation method;
- We propose a novel framework combining distillation and cutting channels, which uses a RPN weighted and hard-soft distribution-wise distillation loss that measures structural differences in teacher-student networks knowledge. As the classification network is a part of object detection network, thus we value the performance of VGG16 network on a simple CIFAR-10 dataset and then value the object detection task on the DIDI-TT dataset on the basis of classification task in Faster-RCNN network [12];
- We show experimentally that our approach provides significant improvements across a variety of experiments and deep network architectures (see section IV), and the improvement rates surpass several popular distillation methods.

II. Related Work

There are three currently used model compression methods. The first one is to change the network's architecture for model compression, such as changing the network layers' number, etc.; the second method is to change the network's weights by quantization method to use low-bit data to compress the model, or express the high-level features with its low-rank features through matrix decomposition; the third method is to merge forward operations to compress the model by merging the Batch Norm layer with previous convolutional layer or fully connected layer to reduce the amount of calculation. In this paper, we focus on the first methods, that is, modifying networks to reduce the model size to reduce the amount of model calculation and model size.

Larger and more complex networks usually have a better performance, but redundant information leads to a large computation calculation and storage operations. The distillation method is to use a large network with a good performance to teach the a small network.

Knowledge distillation was originally proposed by Bucila, Caruana, and Niculescu-Mizil [13], and the main inspiration for this paper is from knowledge distillation [14] by Hinton, Vinyals, and Dean, which compresses the knowledge of a large and computational expensive model to a single computational efficient neural network. Distillation has quickly gained popularity among deep learning and has a variety of applications, e.g., transferring from one architecture to another network, Romero et al.(2014) [15] proposed to transfer knowledge by supervising the difference between teacher and student's intermediate layers.

Knowledge distillation is one approach that transfers knowledge from the teacher model to the student model. FitNet [15] makes the student mimic the full feature maps of the teacher. Czarnecki et al. (2017) [16] minimized the teacher and student derivatives loss and the predictions from teacher model while Tarvainen and Valpola (2017) [17] choose averaging model weights to train the network instead of using predictions from the teacher network. Furlanello et al. [18] and Bagherinezhad et al. [19] demonstrated that by training the student using softmax outputs of the teacher as ground truth over generations. Yim et al. [20] transfers the output activations using Gramian matrices and then fine-tunes the student network.

However, most previous methods only supervise the final part of the teacher and student network. They did not make good use of the network middle part. In this paper, we propose a novel framework combining distillation and cutting channels. We also give an algorithm which uses a RPN weighted and hard-soft distribution-wise distillation loss function to measure structural differences in teacher-student networks knowledge.

III. Method

The purpose of this research is to optimise road traffic sign compression via knowledge distillation on classification and object detection neural networks. The framework of the model is depicted in Fig.1. The classification network is a part of the object detection network, thus we evaluate the performance on a VGG16 network on a light and simple CIFAR-10 dataset and then evaluate the object detection task on the DIDI-TT dataset on the basis of the classification task.

Our work differs from existing approaches in that we first study how to improve the student performance given fixed student and teacher network sizes. Second, by combining several methods, such as cutting channels and layers, modifying RPN network structure, propose teacher-student structural differences etc., and introduces small images samples distillation method in traffic scenarios to improve distillation performance. Our method is based in part of the distillation idea of [21] [14].

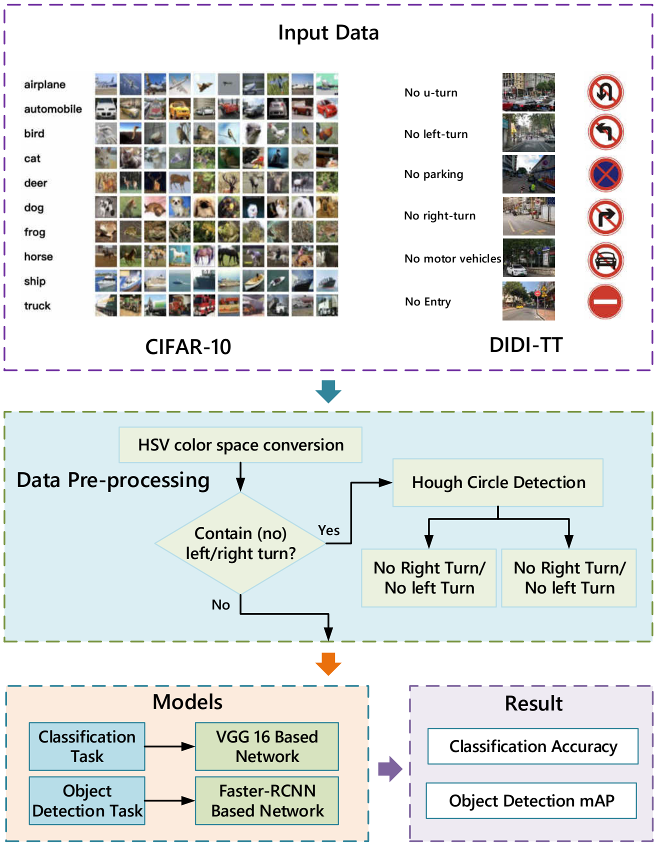

Fig. 1. The framework of the distillation model.

A. Distillation Applied to the VGG16 Classification Network

Distillation methods could be applied to classification networks. This paper designs a simple supervision method, using hard labels and soft labels in the VGG16 network to train the network. The probability value output from the softmax layer of the network trained by the full teacher network (VGG16) is used as the hard label. The probability value output of the softmax layer of the network of the student network training station is used as a soft label after the label distribution is softened in (1). Compared with other logits to convert the logit and z_i of each class to probability q_i, where T is the temperature (and used to soften the model's label distribution) that is normally set to 1. By designing a loss function to supervise the distribution of hard and soft tags, a student network (small network) can be obtained under the supervision of a teaching network. The training framework is shown in Fig.2.

The softening function obtains a soft label by describing any similar structure between classes that need to be labelled. For example, which of the wrong categories is the recognized object more like? The neural network generates a class probability through the output layer of softmax.

$$q_i = \frac{exp(\frac{z_j}{T})}{\sum_j exp(\frac{z_j}{T})}. \tag{1}$$

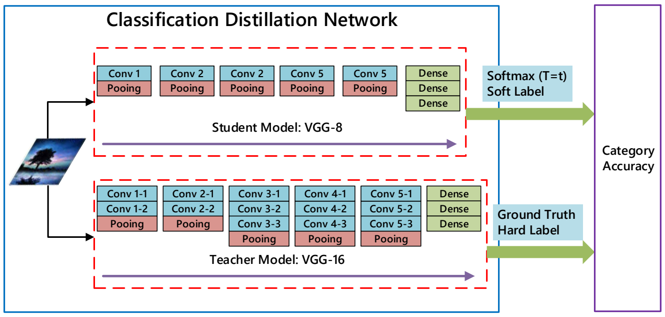

Fig. 2. The architecture of the classification distillation network.

From (1), we see that the larger the T, the softer the soft label distribution is. In the experiment, we tried a variety of T values. In the end, we calculated the sigmoid cross entry to supervise the loss with hard label and soft label. Equation (2) gives the loss function, where y_c represents the variant 0 or 1, y_c assigns 1 when the label of the calculation and sample is consistent, otherwise it assigns 0. p_c is the prediction probability that the sample is classified to label c.

$$L = -\sum_{c=1}^{M} y_c log(p_c). \tag{2}$$

B. Distillation Applied to the Faster-RCNN Detection Network

The training process of our proposed distillation algorithm based on a detection network is implemented based on a Faster-RCNN detection algorithm. Faster-RCNN consists of three modules: 1) shared feature extraction through convolutional layers; 2) a target proposal generation Region Proposal Network (RPN); 3) a Classification and Regression Network (RCN), which returns detection scores and suggested spatial adjustment vectors for each object. Both RCN and RPN use the output of 1) as a feature, and RCN also takes the result of RPN as an input. To achieve highly accurate object detection results, it is critical to learn powerful models for all three components.

Unlike the previous methods [15] [16] that supervise the teacher and student network in the final part of the network only, the distillation method we proposed uses supervision in the middle of the network (in addition to at the end of the network), the training framework is shown in Fig.3. The teacher model of the large network (Faster-RCNN) is initialized by the weighted pre-trained model of the trained detection network. The small network is randomly initialized. When a complete picture is transmitted to the network, it passes through CNNs for teacher and student models to produce two different feature maps. After the feature map is sent to the region of interest (ROI) pooling layer through generating the suggestion box by the RPN, it will generate losses under the supervision of the ground truth label, thereby showing classification and regression results.

The algorithm proposed in this paper is to monitor the smooth L1 loss after the feature map is generated by the

Algorithm 1 Compute the mimic loss to define the loss layer.

```
Require: $self$, $bottom$, $top$ and $propagate\_down$
Ensure: len($bottom$) == 2; $C_{bot1}$ == $C_{bot2}$;
$C_{bot1} \leftarrow$ count of the first bottom
$C_{bot2} \leftarrow$ count of the second bottom
$diff \leftarrow bottom[0] - bottom[1]$
$top[0] \leftarrow \sum_{N=1}^{n} diff_i^2/200/C_{bot1}$
for $i$ in range $(2)$ do
    if $propagate\_down_i$ is False then
        continue
    end if
    if i equals $0$ then
        $sign \leftarrow 1$
    else if i not  then
        $sign \leftarrow 0$
    end if
    $bottom[i]_{diff} \leftarrow sign * diff * top[i]_{diff}/C_{bot1}$
end for
```

CNN for large and small networks and then to calculate the difference L2 [12] loss in the calculation results generated by the ROI pooling layer. Algorithm 1 defines the loss function using pseudocode.

The smoothed L1 loss function is smoother and more robust than the basic L1 loss function. It can converge faster and reduce the probability of gradient explosions. The feature maps calculated by the teacher network and the student CNNs, respectively, are calculated using the smoothed L1 loss function. The average value is calculated. The supervised loss can make the small network more approximate to the structure of the large network in the convolution calculation, thereby transferring the model learning of a large network to a smaller network. The L2 norm loss function is stable, the calculation formula is shown in (3). Among them, y_i represents the network after the small network passes through the ROI layer, and $f(x_i)$ is the network after the large network passes through the ROI layer function.

$$S = \sum_{i=1}^{n} |Y_i - f(x_i)|. \tag{3}$$

By reducing the two weighted losses, the small network is trained. We first calculate the loss of teachers and students networks separately, and then combine the two losses into one loss for optimization, and end-to-end update can obtain better accuracy. In the process, the intermediate results of the teacher network are learned step by step, and finally the target detection results are output through the fully connected layer. In subsequent experiments, we can obtain better distillation model parameters by adjusting the ratio of the two losses.

IV. Experiments

In this section, we use three real datasets to conduct the experiments and perform the distillation on the classification network and detection network separately to verify the effectiveness of distillation on different kind of networks. All the datasets are divided into a training set, validation set and test set by the dataset publishers.

TABLE II
Traffic Sign Dataset Illustration

Type	**Name**	**Sample**
No u-turn	p5	
No motor vehicles	p10	
No right turn	p19	
No left turn	p23	
No parking	pn	
No entry	pne	

TABLE III
A Summary of DIDI-TT Dataset Attributes

Attribute	**Description**
Light	Record different lighting scenes
Record method	Hang the signs in a row, record 6 signs at a time, then cut out each one
Number	500 images per hour
Time	images recorded and a total 60000 images from 6 a.m. to 6 p.m.
Angle	Left-right: -90° +90° Up-down: -90° +90° Record different angles of the sphere

A. Dataset Description

1) CIFAR-10 dataset: The CIFAR-10 dataset [22] consists of 32 × 32 RGB images. The task for the dataset is to classify images into 10 image categories. CIFAR-10 contains 10 classes. This dataset is used in the classification distillation experiments.

2) VOC dataset: PASCAL VOC 2007 [23] is a relatively small dataset that contain less object categories and labeled images, which suits traffic scenarios. We have done several experiments on this dataset to validate our proposed distillation method and for comparison with other methods.

3) DIDI-TT dataset: The DIDI-TT dataset used in this research contains generic traffic signs collected from different lighting conditions and camera angles. The dataset was taken from 1 November to 1 December in Haidian district, Beijing city, China, and mainly came from mobile terminals including Huawei Honor, Xiaomi 5, Samsung S7e devices. The types and the collection attributes are shown in Table II. It concludes six categories with all location information of the image. The DIDI-TT data set is made into a VOC data format, where the attribute values are described as follows in Table III.

B. Distillation Applied to the VGG16 Classification Network

1) VGG16 Classification Network: The VGG16 network is a simple network focusing on building convolutional layers which does not have too many hyperparameters. First, a 3 ×

3 filter with a stride of 1 is used to construct the convolution layer, and the padding parameter is a parameter in the same convolution. Then a 2 × 2 filter with a stride of 2 is used to build the maximum pooling layer.

2) Baseline Distillation Experiment: Distillation classification experiments were performed on VGG16 using the CIFAR-10 dataset. Then we simply modified the network structure of VGG16, i.e., we used the original VGG16 network and the modified 8-layer VGG network, which is implemented by a part of the convolution layers in VGG16. The VGG8 network architecture is: conv1, conv2, conv3, conv4, conv5. The fully connected layers and channels remain constant, Table IV shows the experiment result that the accuracy of large network and small network is 78.2% and 75.7% respectively.

TABLE IV
RESULTS OF THE BASELINE CLASSIFICATION EXPERIMENTS WITH 60000 ITERATIONS

Name	Iteration	Loss	Accuracy
Train_VGG16	60000	0.739256	78.20%
Train_VGG8	60000	0.893016	75.70%

3) Distillation Experiment for Different Channel Number and Learning Rate Reduction Strategies: After only 60,000 iterations of the network, the effect is not ideal, so a method of continuous training is adopted for the network. The results were obtained under different initialization methods, channel number and learning rate strategies. Table V shows that as the amount of calculation decreases (the number of layers, the number of channels), the accuracy rate decreases. There is no significant difference in the impact of the different learning rate reduction strategies on the results.

TABLE V
RESULTS OF THE BASELINE CLASSIFICATION EXPERIMENTS FOR SEVERAL TRAINING CONDITIONS

Name	Initialization parameters	Channel	Lr policy	Loss	Accuracy
VGG16	conv xavier fc xavier	1	-	-	89%
VGG8	conv xavier fc gaussian	1	step	0.51	86.92%
VGG8		1	step	0.76	85.23%
VGG8		1/2	step	0.82	81.14%
VGG8		1/4	poly	0.88	81.15%
VGG8		1/4	poly	0.8	77.49%
VGG8		1/8	poly	0.86	73.20%

4) Distillation Experiment for Different Temperatures, Loss Functions Types and Ratios: Distillation experiments were performed in the CIFAR-10 dataset. The large network is a complete VGG16 network, the small network is a VGG8 network, and the number of channels is set to 1/8 of the original network. The experiment results at different temperatures, loss function types, and the accuracy of the soft label to hard label ratio. Table VI shows some results with different loss function, temperatures and ratio.

TABLE VI
RESULTS OF THE DISTILLATION CLASSIFICATION EXPERIMENTS

No.	Loss function	Temperature	Ratio (hard-soft with temperature)	Accuracy
1	sigmoidcross	10	0.7*10*10 / 0.3	73.30%
2	sigmoidcross	10	0.5*10*10 / 0.5	83.40%
3	sigmoidcross	10	0.7 / 0.3	75.60%
4	sigmoidcross	10	0.3 / 0.7	71.40%
5	sigmoidcross	10	0.3*10*10 / 0.7	72.10%
6	L2	10	0.7*10*10 / 0.3	76.10%
7	L2	10	0.7 / 0.3	74.80%
8	Sigmoidcross	20	0.7 / 0.3	83.50%
9	Sigmoidcross	20	0.7*20*20 / 0.3	**84.20%**
10	sigmoidcross	50	0.7 / 0.3	73%
11	sigmoidcross	50	0.7*50*50 / 0.3	73.40%
12	sigmoidcross	5	0.7*5*5 / 0.3	72.80%
13	sigmoidcross	5	0.7 / 0.3	73.90%
14	sigmoidcross	5	0.7*10*10 / 0.3	73.30%
15	sigmoidcross	1	0.7 / 0.3	83.90%
16	sigmoidcross	1	0.5 / 0.5	73.90%
17	sigmoidcross	2	0.7 / 0.3	73.70%
18	sigmoidcross	2	0.7*2*2 / 0.3	74.70%
19	sigmoidcross	2	0.7 / 0.3	73.30%
20	sigmoidcross	/	0.7 / 0.3	83.40%

Experiments number 1-7 focused on the effect of distillation at a temperature of 10, and found that using a sigmoidcross [24] loss function under the same ratio conditions gave better results. Experiments number 8 and 9 found if that when the same temperature and loss function are squared, better results can be obtained. Other experiments have found that using a temperature of 20 can get the best results. Comparing different ratios [25], it is found that the hard target uses a larger ratio and that the hard target and soft target ratio is 0.7:0.3.

5) Classification Distillation Results Discussion: From the classification we can see the baselines of the large network (VGG16) and small network (VGG8-conv1/8) are 86.91% and 73.2%, respectively. The learning ability of the small network can be increased to 84.2% by distillation, which is equivalent to a case where the model parameters are reduced by about 16. Next, the accuracy rate has dropped by only 2%, which is 11% higher than the accuracy rate of the small network itself. This shows that distillation is very effective in image classification tasks.

We compare different strategies for classification distillation to highlight the effectiveness of our proposed framework. We choose VGG16 as the teacher model and channel cut VGG8 as our student model. We can conclude that the classification distillation using our proposed method can lose the least model information, that is, the least reduction in accuracy. A clear reduction in training model and accurate percentage of the model parameters is shown in Table VII.

TABLE VII
CLASSIFICATION DISTILLATION COMPARISON OF THE NEW METHOD AGAINST THE BASELINE

Name	Teacher Model	Student Model	Accuracy Decrease	Parameter Decrease
Ours	VGG16 86.91%	VGG8/8 84.20%	**-2.71%**	-43.77%
	VGG16 86.91%	VGG8/4 73.20%	-13.71%	-46.88%
Mutual [26]	WRN-28-10 78.69%	ResNet-32 69.48%	-9.21%	**-48.63%**
	MobileNet 73.65%	ResNet-32 69.12%	-4.53%	-34.85%

C. Distillation Applied to the Faster-RCNN Detection Network

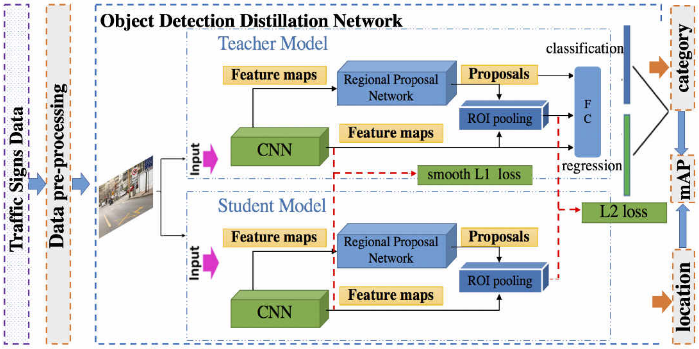

Fig. 3. The framework of the detection distillation model.

1) Faster-RCNN Detection Network: In this part, the distillation algorithm has been applied to a Faster-RCNN based object detection network. The front end of the RCNN network is a classification network. The object detection distillation network was trained by supervising the loss of the rear output of large networks and small networks that reduce the number of channels and layers. The distillation algorithm cuts off 1/2, 1/4, 1/8 of the number of channels of the network. The number of channels in the last layer conv 5-3 does not change and is kept at 512. All experiments were conducted with 70000 rounds of training on the DIDI-TT dataset.

2) Baseline Experiment Distillation: Fig.4 shows the baseline experiment for models that change the network structure, the learning rate needs to be adjusted to be non-zero, the initialization strategy in CNN layer and FC layer is Xavier and Gaussian respectively. We found that the results of each network did not perform well without using a pre-trained model.

In this experiment, S1, S4, S6-8 use the pre-trained model, S3 uses the model that we trained for 70000 iterations, and S2 doesn't use a pre-trained model. The channel of student network in S3-5 is half of the other experiments, e.g., S6-8, and the learning rate of S4 is adjusted as 1 and 2. S6-8 represent the networks which have been activated and calculated. S6 is student convolution and backend that is

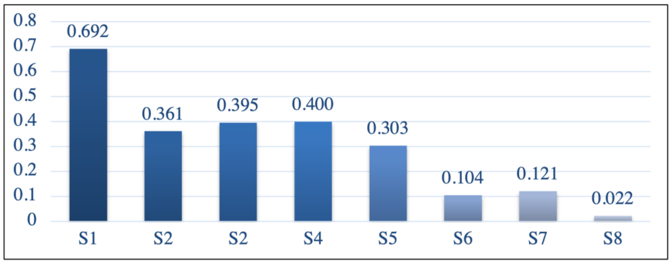

Fig. 4. Object detection baseline results for several training conditions.

TABLE VIII
NEW METHOD VERSUS BASELINE COMPARISON EXPERIMENT USING THE TT100K DATASET

Network	Theretical computation	Accuracy	Benchmark	Fine-grained accuracy	
mobilenet v1 prune_75	0.534G	0.4322	1.42s	p5	57.57%
				p10	21.79%
				p19	36.88%
				p23	27.30%
				pn	93.47%
				pne	22.31%
mobilenet v1 prune_0.5	1.44G	0.4673	1.82s	p5	58.18%
				p10	21.73%
				p19	32.31%
				p23	33.95%
				pn	94.76%
				pne	39.48%
mobilenet v1	2.88G	0.6029	3.64s	p5	76.42%
				p10	35.43%
				p19	53.30%
				p23	53.39%
				pn	93.64%
				pne	49.57%
mobilenet v2	1.53G	0.6956	3.40s	p5	80.30%
				p10	58.36%
				p19	64.53%
				p23	64.53%
				pn	93.23%
				pne	56.44%

TABLE IX
OBJECT DETECTION DISTILLATION OVERALL RESULTS

Name	Model	Original Model	mAP
Base-Pretraining	Teacher	VGG16	68.05%
Base	Teacher	VGG16	39.51%
Distillation	Student	VGG16/2	64.40%
Distillation	Student	VGG16/4	58.00%
Distillation	Student	VGG16/8	47.77%

TABLE X
OBJECT DETECTION DISTILLATION PARAMETER COMPARISON

Attributes	**Original model**	**Distillated model**	**Optimized percent**
Theoretical Computation	948M	431M	-54%
Process speed	2.1s/picture	1.2s/picture	-43%
Parameter number	24M	12M	-50%
mAP	68.05%	64.40%	-5.40%

trained. S7 is the distillation network and teacher's network backend that is trained. S8 stands for distillation network and student's network backend that is trained. As a supplementary verification experiment, we selected different networks to compare theoretical calculations, accuracy, and fine-grained performance of the six selected categories, the comparison experiment is shown in Table VIII.

3) Proposed Different Loss Weight and Channel Distillation Experiment: We compared the distillation mAP under different loss weight and different channel number in Fig.5. mAP is computed from the average precision over all classes,

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i. \tag{4}$$

where C is the total classes number of the objects and AP_i is the $i-th$ class AP value. Through horizontal comparison, it is found that as the number of channels decreases, the performance of the model decreases. RF indicates that the $ROI-pooling+full$ connection layer is initialized, and $T+RF$ indicates that the $teacher+ROI-pooling+full$ connection is initialized. After joining the teacher network, all the mAP increase are better than 10% compared with student networks, indicating the benefits of using distillation.

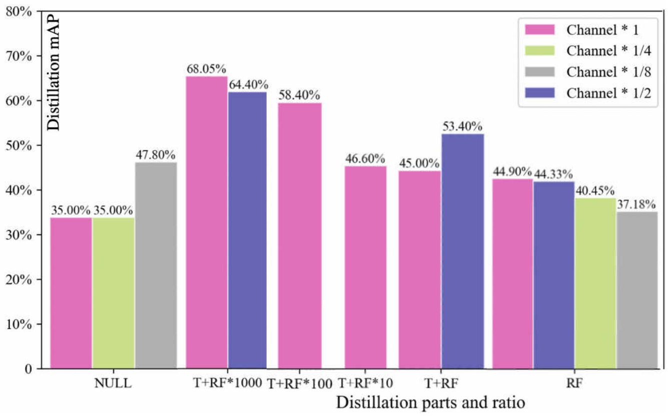

Fig. 5. Loss weight and channel distillation experiments result.

4) Detection Distillation Results Discussion: In the detection task, we finally obtained a comparison by trying multiple initialization methods, multiple small network layer attempts, multiple loss position designs, multiple loss weight designs, and multiple model initialization positions. We achieved good experimental results, that is, by using existing loss joins, ROI-pooling can effectively compress network parameters and achieve better model results. The Table IX below uses the parameters of each model. The network results after reducing the parameters and adding distillation are better than not using the pre-trained model. In the distillation experiment, the baselines of the large network (VGG16) and small network (VGG16/8) are 47.9% and 37.18%, respectively. The learning ability of the small network can be increased to 47.8% by distillation, which is equivalent to reducing the model parameters by about 8% under the circumstances, the accuracy of mAP (mean average precision) is improved by about 10% compared with the small network itself, and it is almost the same as the mAP size of the large network that does not apply the pre-trained model, which indicates that distillation has successfully pre-trained for the detection task. The model is transferred to other smaller networks, which proves the idea of transfer learning and also proves that distillation is also effective for detection tasks. Table X shows the parameter comparison of a base VGG16 model with pre-training against the VGG16/2 distillation model. Table XI shows mAP and the increase for several teacher-student model pairs on VOC object detection database. We compare different strategies for distillation. Our method selects VGG16 without a pretrained model as the teacher model and cut the channel of VGG8 and VGG16 as the student model. We could find that the distillation model mAP surpasses the teacher model. Other choices reflect similar trends. The blank square means that only the teacher model participate in the calculation.

TABLE XI
OBJECT DETECTION DISTILLATION COMPARISON ON THE VOC DATASET

Name	**Teacher Model**	**Student Model**	**mAP (%)**	**mAP (%) Increase**
Ours	VGG16 trained	-	68.05	**/**
	VGG16	-	**39.51**	/
	VGG16	VGG8/4	**64.4**	**24.89**
	VGG16	VGG16/4	**58**	**18.49**
	VGG16	VGG16/8	**47.77**	**8.26**
Fine-grained [27]	Res101	-	74.4	**/**
	Res101h		**67.4**	/
	Res101h	Res101h-I	**71.2**	**3.8**
	VGG16	-	70.4	**/**
	VGG11		**59.6**	/
	VGG11	VGG11-I	**67.6**	**8**
	Res101	-	74.4	**/**
	Res50		**69.1**	/
	Res50	Res50-I	**72**	2.9
Efficient [21]	Tucker	-	**54.7**	/
	Tucker	AlexNet	**57.6**	2.9
	Tucker	VGGM	**58.2**	3.5
	Tucker	VGG16	**59.4**	**4.7**
	AlexNet	-	**57.2**	/
	AlexNet	VGGM	**59.2**	2
	AlexNet	VGG16	**60.1**	**4.7**
	VGGM	-	**59.8**	/
	VGGM	VGG16	**63.7**	2.9

V. Conclusions and Further work

In this paper, we have proposed a novel framework for classification and object detection distillation tasks which are separately based on VGG16 network and Faster-RCNN network, which combines distillation and cutting channels and a Region Proposal Network (RPN) weighted and hard-soft distribution-wise distillation loss that measure structural differences in teacher-student networks knowledge, and this method is useful to reduce parameters while get efficient models. Demonstrating the knowledge distillation on VGG16 as the backbone of Faster-RCNN network, we conduct learning loss of the student and teacher network (after ROI pooling), it is obvious that there are improvements over different hyper-parameters experiments in both the classification and object detection tasks.

In traffic sign identification scenarios, smaller size of models are useful when applied to aid vehicle navigation in real-time situations. We apply the distillation algorithm to experiments using real-world datasets, and perform a series of processing on small target images of traffic signs. Compared with previous distillation methods [21] [27], our distillation algorithm is superior to other algorithms in terms of performance. We find that the distillation algorithm has obvious positive parameter reduction effects and an increased accuracy for classification problems and detection problems, thus it can be used to support the transfer of learning between different size CNNs [20]. A direction for future work is to increase the performance in the classification and object detection tasks, which could also be deployed to various learning schemes, such as auto machine learning [28], reinforcement learning [29]. In this work we combined part of a pruning method with distillation, which could also be expected to integrate the compression methods via both knowledge distillation and other compressing techniques, such as network quantization.

Acknowledgment

This research was funded in part by a Ph.D. scholarship funded jointly by the China Scholarship Council and QMUL. We thank our colleagues from Didi Chuxing who provided insight and expertise that greatly assisted the research.

References

[1] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[2] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," in *Applied Intelligence*. Springer, 2021, pp. 1–30.

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[4] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive bayes classification," *Information Sciences*, vol. 179, no. 19, pp. 3218–3229, 2009.

[5] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

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

[7] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using googlenet," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 89–96.

[8] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2017, pp. 783–787.

[9] A. Balanskat, "The ict impact report: A review of studies of ict impact on schools in europe, european schoolnet," *http://insight. eun. org/shared/data/pdf/impact_study. pdf*, 2006.

[10] C. Tai, T. Xiao, Y. Zhang, X. Wang *et al.*, "Convolutional neural networks with low-rank regularization," *arXiv preprint arXiv:1511.06067*, 2015.

[11] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

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

[13] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[16] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4278–4287.

[17] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[18] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," *arXiv preprint arXiv:1805.04770*, 2018.

[19] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, "Newtonian scene understanding: Unfolding the dynamics of objects in static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3521–3529.

[20] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V. I. Morariu, X. Han, M. Gao, C.-Y. Lin, and L. S. Davis, "Nisp: Pruning networks using neuron importance score propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9194–9203.

[21] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[22] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[23] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes homepage," 2015.

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[25] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale l2-loss linear support vector machines," *Journal of Machine Learning Research*, vol. 9, no. Jul, pp. 1369–1398, 2008.

[26] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[27] T. Wang, L. Yuan, X. Zhang, and J. Feng, "Distilling object detectors with fine-grained feature imitation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4933–4942.

[28] V. Macko, C. Weill, H. Mazzawi, and J. Gonzalvo, "Improving neural architecture search image classifiers via ensemble learning," *arXiv preprint arXiv:1903.06236*, 2019.

[29] A. Ashok, N. Rhinehart, F. Beainy, and K. M. Kitani, "N2n learning: Network to network compression via policy gradient reinforcement learning," *arXiv preprint arXiv:1709.06030*, 2017.