Model Distillation with Knowledge Transfer in Face Classification, Alignment and Verification

Chong Wang and Xipeng Lan*
Beijing Orion Star Technology Co., Ltd.
Beijing, China
chongwang.nlpr@gmail.com, xipeng.lan@gmail.com

Abstract

Knowledge distillation is a potential solution for model compression. The idea is to make a small student model imitate the output of a large teacher model, thus the student that is competitive to the teacher can be obtained. Most previous studies focus only on the classification task where they propose different teacher supervision, but other tasks are barely considered, and they mostly ignore the importance of student initialization. To overcome the two limitations, in this paper, we propose face model distillation with strong student initialization and knowledge transfer, which can boost not only the task of face classification, but also domain-similar tasks including face alignment and verification. First, in face classification, a student model with all layers initialized is trained in a multi-task way with its class labels and teacher supervision. Then, the similar multi-task training is adopted with the knowledge transferred from classification to alignment and verification. Evaluation on the CASIA-WebFace and CelebA datasets demonstrates that the student can be competitive to the teacher in all the three tasks, and even surpasses the teacher under appropriate compression rates. Moreover, we also test the proposed method on the large-scale MS-Celeb-1M database, where the student can also achieve competitive performance.

1 Introduction

Since the emergence of Alexnet[23], larger and deeper networks have shown to be more powerful[36, 37, 16]. However, as the network going larger and deeper, it becomes more difficult to use it in embedding systems or mobile devices. Therefore, model compression has become necessary in compressing the large network into a small one. In recent years, many compression methods have been proposed, including knowledge distillation[29, 3, 1, 41, 27, 34, 18, 33], weight quantization[11, 7, 31], weight pruning[15, 13, 14, 20, 38], weight decomposition[46, 4, 9, 39, 28] and selective execution[40, 25, 10]. In this paper, we focus on the knowledge distillation, which is a potential approach for model compression.

In knowledge distillation, there is usually a large teacher model and a small student one, and the objective is to make the student competitive to the teacher by imitating the teacher's supervision, which is usually the output of the teacher network. Previous studies mainly consider how to select the best teacher supervision, *e.g.,* the hidden layer[27], logits[1, 41, 34] or soft predictions[18, 33], but they barely realize the importance of student initialization[33]. More importantly, they consider only the classification task on datasets like CIFAR[22], which limits the application of knowledge distillation in various tasks. In this paper, we take face recognition as a breaking point, and consider model distillation on domain-similar tasks including face classification, alignment and verification, and also its application on large-scale datasets[44, 26, 12].

*Chong Wang and Xipeng Lan have equal contribution.

For the model distillation on non-classification tasks, some hints have been given. The tasks of object detection[32], object segmentation[5], image retrieval[47] and face verification[30] have used the pretrained classification models as initialization[23, 36, 16], which gives us strong implication that the distilled knowledge can also be transferred to other tasks in a similar way. The availability of this idea mostly comes from the fact that the domains of these tasks are similar, which makes them share a lot in the network from low-level to high-level representation[45]. Motivated by the fact, we transfer the distilled knowledge by taking the classification models as initialization for training new models in face alignment and verification.

For the model distillation in classification, initialization is important. It is shown in [17, 19, 2, 36, 45] that training deep networks can be assisted by layer-wise initialization and intermediate supervision[37]. Based on their advantages, Fitnet[33] initializes student's shallow layers by regressing teacher's intermediate supervision. However, as demonstrated in [1, 41] that deeper student networks can give better distillation, initializing only the shallow layers is hard to learn teacher's high-level representation that mostly comes from deep layers. Besides, the network transferability increases as tasks are more similar[45], which indicates that we should initialize more deep layers as we focus only on the classification task. Based on both evidence, initializing the student with all layers is more reasonable, which can give stronger transferability for high-level representation, thus training competitive student networks can be easier.

In this paper, we focus on the face recognition problem, and propose strong student initialization and knowledge transfer for model distillation, which can be easily used in some domain-similar tasks, *e.g.*, face classification, alignment and verification. First, in classification, we initialize all the layers of the student network with 0/1 class labels, and enhance it with a multi-task way which uses the class labels and teacher supervision jointly. Then, to train the new teachers and students in alignment and verification, we transfer the distilled knowledge by taking the teacher and student models in classification as initialization, and the knowledge distillation is processed in a similar multi-task way. Experiments on the CASIA-WebFace[44] and CelebA datasets[26] show that the student initialization and knowledge transfer can largely improve the distillation performance in all the three tasks, where the student can be competitive to the teacher, and even exceeds the teacher under appropriate compression rates. Besides, to validate our method on large-scale face classification problem, we test it on the challenging MS-Celeb-1M database[12] (100 thousand people with 9 million images), where the student also achieves encouraging performance.

2 Related work

In this part, we introduce some previous studies on knowledge distillation. Buciluǎ *et al.*[3] propose to generate synthetic data by a teacher network, and a student is trained with the data to mimic the 0/1 class labels of the teacher. However, Ba and Caruana[1] observe that the 0/1 labels have lost teacher's uncertainties which are more informative, thus they propose to regress the logits (pre-softmax activations)[18]. Besides, they prefer the student to be deep, which is good to mimic complex functions. To better learn the function, Gregor *et al.*[41] observe the student network should not only be deep, but also be convolutional, and they get competitive performance to the teacher in CIFAR[22]. Most methods need a large ensemble of teachers for supervision, but this will take a long training time and runtime[34]. To address the issue, Sau and Balasubramanian[34] propose noise-based regularization that can simulate the logits of multiple teachers. However, Luo *et al.*[27] observe the values of the logits are unconstrained, and its high dimensionality will also cause fitting problem. Thus they use the top hidden layer, as it captures as much information as the logits but more compact. All these methods only use single teacher supervision, and if it fails, the training will be difficult. To solve the problem, Hinton *et al.*[18] propose a multi-task approach which uses 0/1 class labels and teacher supervision jointly. Particularly, they use the post-softmax activations with temperature smoothing as the teacher supervision, which can better represent the label distribution. One problem is that student networks are mostly trained from scratch without considering the importance of student initialization[33]. Base on this, Romero *et al.*[33] propose to initialize the shallow layers of the student by regressing the teacher's mid-layer output. However, previous studies consider only the classification task on datasets like CIFAR[22], which largely limits the application of knowledge distillation in various vision tasks. Besides, the initialization of the student's shallow layers cannot make it easy to learn the teacher's high-level representation in deep layers. In this paper, we explore the solutions in face recognition problems.

3 Method

In this part, we elaborate the face model distillation with student initialization and knowledge transfer, which can be applied in face classification, alignment and verification. We first review the idea of knowledge distillation, and then introduce how to initialize the student network in classification and transfer the distilled knowledge to other tasks.

3.1 Review of knowledge distillation

We adopt the multi-task based distillation framework in Hinton *et al.*[18], which is summarized as follows. Let T and S be the teacher and student network, and their post-softmax predictions to be $\mathbf{P_T} = \mathrm{softmax}(\mathbf{a_T})$ and $\mathbf{P_S}$=$\mathrm{softmax}(\mathbf{a_S})$, where $\mathbf{a_T}$ and $\mathbf{a_S}$ are the pre-softmax predictions, also called the logits[1, 41, 34]. However, the post-softmax predictions have lost some relative uncertainties which are more informative, thus a temperature parameter τ is used to smooth predictions $\mathbf{P_T}$ and $\mathbf{P_S}$ to be $\mathbf{P_T^\tau}$ and $\mathbf{P_S^\tau}$, which are denoted as ***soft predictions***:

$$\mathbf{P_T^\tau} = \mathrm{softmax}(\mathbf{a_T}/\tau), \quad \mathbf{P_S^\tau} = \mathrm{softmax}(\mathbf{a_S}/\tau). \tag{1}$$

Then, the objective is to optimize the following loss function

$$\mathrm{L}(\mathbf{W_S^{cls}}) = \mathrm{H}(\mathbf{P_S}, \mathbf{y}) + \lambda \mathrm{H}(\mathbf{P_S^\tau}, \mathbf{P_T^\tau}), \tag{2}$$

wherein $\mathbf{W_S^{cls}}$ is the network parameters of the student network, and $\mathbf{y}$ is the 0/1 class labels. For simplicity, we omit min and the number of samples N in all of our loss functions, and denote the upper right symbol cls as the network parameters in classification. $\mathrm{H}(,)$ is the cross-entropy, thus the first term is the usual softmax loss, while the second one is the cross-entropy between the soft predictions of the teacher and student network, with λ balancing between the two terms. This multi-task training is advantageous over single teacher supervision because the teacher cannot be guaranteed to be always correct, and if fails, the class labels will take over the training.

3.2 Student initialization

In this part, we elaborate how to improve knowledge distillation in classification by initializing students. One question we first have to think about is why we need initialization in model distillation. In usual classification tasks, initialization is mainly used to alleviate the gradient vanishing problem[19, 2, 6, 17]. Recently, networks can be trained easier with batch normalization[21], and it seems initialization is no longer an issue. Compared to the usual classification, model distillation is a generalized classification task with additional teacher supervision, which is the key factor to improve the student to a higher level. Intuitively, it will be easier for teachers to teach students with much prior knowledge, which is the student initialization. Therefore, the initialization in model distillation focuses on giving a better student, but not the gradient vanishing problem.

As analyzed above, knowledge distillation can be summarized into two steps: initialization and learning from teachers. Hinton's framework[18] in Sec.3.1 considers no initialization, and learns with teachers from scratch; while Fitnet[33] first initializes student's shallow layers by regressing the teacher's mid-layer output, then it follows Eqn.(2) for fine-tuning. But as shown in [1, 41] that deeper student networks can give better distillation, only initializing the shallow layers is difficult to learn teacher's high-level representation, which is highly-correlated to deep layers. Furthermore, [45] demonstrates that the network transferability increases as tasks become more similar, while the tasks of initialization and learning from teachers are closely related, thus more deep layers should be initialized. Base on both evidence, initializing student networks with all layers can give stronger transferability, thus it is easier to train competitive students.

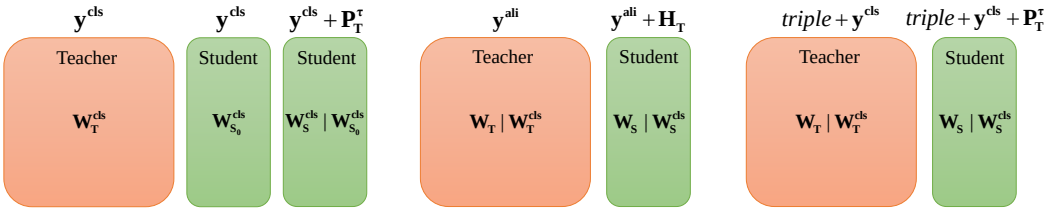
To obtain a student network as initialization, we use the usual softmax loss with 0/1 class labels:

$$\mathrm{L}(\mathbf{W_{S_0}^{cls}}) = \mathrm{H}(\mathbf{P_S}, \mathbf{y}), \tag{3}$$

wherein the lower right symbol S_0 denotes the initialization for student network S. Then, we improve Eqn.(2) with the following loss function

$$\mathrm{L}(\mathbf{W_S^{cls}}|\mathbf{W_{S_0}^{cls}}) = \mathrm{H}(\mathbf{P_S}, \mathbf{y}) + \lambda \mathrm{H}(\mathbf{P_S^\tau}, \mathbf{P_T^\tau}), \tag{4}$$

wherein $\mathbf{W_S^{cls}}|\mathbf{W_{S_0}^{cls}}$ indicates that $\mathbf{W_S^{cls}}$ is trained with the initialization of $\mathbf{W_{S_0}^{cls}}$, and the two entropy terms remain the same to Eqn.(2). This process is shown in Fig.1(a).



(a) Distillation: classification (b) Distillation: alignment (c) Distillation: verification

Figure 1: The pipeline of model distillation in face classification, alignment and verification. $\mathbf{y}^{\mathbf{cls}}$ and $\mathbf{y}^{\mathbf{ali}}$ are the true labels for classification and alignment respectively.

3.3 Knowledge transfer

In this part, we show how to transfer the distilled knowledge from face classification to face alignment and verification. To transfer the knowledge, many studies[32, 5, 47, 30] have used pretrained classification models as initialization[23, 36, 16], and the similar domain makes the classification network transfer easily to other tasks[45]. Inspired by them, we use a similar way by taking classification models as initialization for transfer.

It is easy to transfer the student by taking $\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}$ as initialization, but only transferring the student is not enough. We also have to transfer the teacher because the soft predictions in classification may not fit well in other tasks, *e.g.,* teachers for mathematics and machine learning are different, although mathematics is widely used in machine learning. One problem is different tasks have different loss functions which lead to different teacher supervision, thus how to select the best supervision remains discussion. In the following, we discuss this problem for alignment and verification. For convenience, we denote the network parameters of the teacher in classification to be $\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}$.

3.3.1 Alignment

Without loss of generality, there is no any class labels in alignment, but only the keypoint locations for each image. Face alignment is usually considered as a regression problem, and we train the teacher network with optimizing the Euclidean loss:

$$\mathrm{L}(\mathbf{W}_{\mathbf{T}}|\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}) = \|\mathbf{R}_{\mathbf{T}} - \mathbf{y}\|^2, \tag{5}$$

wherein $\mathbf{R}_{\mathbf{T}}$ and $\mathbf{y}$ are the regression prediction and target. Given the alignment teacher $\mathbf{W}_{\mathbf{T}}$ and the classification teacher $\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}$, which supervision do we learn from? One choice is the top hidden layer generated by $\mathbf{W}_{\mathbf{T}}$, and it can describe high-level representation; while another is the soft predictions $\mathbf{P}_{\mathbf{T}}^{\tau}$ generated by $\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}$, which can provide identity information. Model distillation with the two supervision can be optimized with the loss functions shown in Eqn.(6) and Eqn.(7). $\mathbf{H}_{\mathbf{S}}$ and $\mathbf{H}_{\mathbf{T}}$ are the representation of the top hidden layer, and they satisfy $\mathbf{R}_{\mathbf{S}} = fc(\mathbf{H}_{\mathbf{S}})$ and $\mathbf{R}_{\mathbf{T}} = fc(\mathbf{H}_{\mathbf{T}})$, where fc denotes the fully-connected mapping.

$$\mathrm{L}(\mathbf{W}_{\mathbf{S}}|\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}) = \|\mathbf{R}_{\mathbf{S}} - \mathbf{y}\|^2 + \lambda\|\mathbf{H}_{\mathbf{S}} - \mathbf{H}_{\mathbf{T}}\|^2, \tag{6}$$

$$\mathrm{L}(\mathbf{W}_{\mathbf{S}}|\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}) = \|\mathbf{R}_{\mathbf{S}} - \mathbf{y}\|^2 + \lambda\mathrm{H}(\mathbf{P}_{\mathbf{S}}^{\tau}, \mathbf{P}_{\mathbf{T}}^{\tau}), \tag{7}$$

It is not necessary to discuss which one is better, but to choose the appropriate one. Consider when we use the soft prediction $\mathbf{P}_{\mathbf{T}}^{\tau}$ in alignment. It can provide the identity information that an image in the alignment dataset looks like which identity in the classification dataset. However, as mentioned above, there is no any class labels in the alignment task, thus the identity information in classification and alignment will be different. If the number of identity in classification is not extremely large, telling an image in the alignment set looks like which identity in the classification set is not meaningful, and it will even harm the alignment performance when the difference in region and race is too large. However, if the hidden layer is used, what we need to learn is only a linear mapping $\mathbf{R}_{\mathbf{S}} = fc(\mathbf{H}_{\mathbf{S}})$, which can be easier and meaningful as supervision for the alignment task. Therefore, in face alignment, the top hidden layer seems like more appropriate, and Eqn.(6) is adopted for model distillation. Fig.1(b) illustrates the distillation of alignment.

3.3.2 Verification

The task of face verification is to determine if two images belong to the same identity. Triplet loss[30, 35] is a widely studied topic in verification[35], and we take it for model distillation. Particularly, in training the triplet model, we have the same 0/1 class labels as in classification[44, 12], then the teacher network trained with triplet loss can be optimized as

$$\mathrm{L}(\mathbf{W_T}|\mathbf{W_T^{cls}}) = \left[\|\mathbf{H_T^a} - \mathbf{H_T^p}\|^2 - \|\mathbf{H_T^a} - \mathbf{H_T^n}\|^2 + \alpha\right]_+ + \lambda \mathrm{H}(\mathbf{P_T}, \mathbf{y}), \tag{8}$$

where $\mathbf{H_T^a}$, $\mathbf{H_T^p}$ and $\mathbf{H_T^n}$ are the top hidden layers for the anchor, positive and negative samples respectively, *i.e.,* a and p belong to the same class, while a and n come from different classes. Besides, α controls the margin between positive and negative pairs. If $\lambda = 0$, Eqn.(8) turns into the traditional triplet loss without considering any class labels; while researchers observe the class labels can be beneficial to enhance the triplet loss[24], wherein $\lambda > 0$ is set.

Similar to the alignment task, we consider the top hidden layer and soft predictions as two possible teacher supervision in model distillation. In training the triplet loss, one major problem is the training will be easily over-fitting as the hidden layer has to strictly satisfy the margin, which will harm the soft predictions learned in classification. We clearly observe this in our experiments, and the over-fitting becomes even more serious with online hard negative mining[35]. Although additional class labels can mostly alleviate this problem, without setting an appropriate margin, the Euclidean distance and soft predictions are still contradictory. However, if we use the soft prediction $\mathbf{P_T^\tau}$ as supervision, due to the same class labels as in classification, $\mathbf{P_T^\tau}$ is beneficial to retain the identity information, and the additional class labels will further enhance the power of $\mathbf{P_T^\tau}$, as shown in Fig.1(c). Therefore, in the verification task, the teacher supervision with soft predictions is more appropriate, and the loss function of distillation turns into

$$\mathrm{L}(\mathbf{W_S}|\mathbf{W_S^{cls}}) = \left[\|\mathbf{H_S^a} - \mathbf{H_S^p}\|^2 - \|\mathbf{H_S^a} - \mathbf{H_S^n}\|^2 + \alpha\right]_+ + \lambda_1 \mathrm{H}(\mathbf{P_S}, \mathbf{y}) + \lambda_2 \mathrm{H}(\mathbf{P_S^\tau}, \mathbf{P_T^\tau}). \tag{9}$$

4 Experimental evaluation

In this section, we give the experimental evaluation of the proposed method. We first introduce the experimental setup in detail, and then show the results of model distillation in the tasks of face classification, alignment and verification.

4.1 Experimental setup

Database: We use three popular datasets for evaluation, including CASIA-WebFace[44], CelebA[26] and MS-Celeb-1M[12]. CASIA-WebFace is a mid-scale classification database, which contains 10575 people and 494414 images in total. CelebA is a large-scale database for face alignment, and there are 10177 people with 202599 images. However, the identity labels are not public, and only the five keypoint locations are available. Compared to the previous two, MS-Celeb-1M is a large-scale classification dataset, which contains 100 thousand people with almost 9 million images in total. In experiments, we use CASIA-WebFace for face classification and verification, CelebA for face alignment, and the large-scale MS-Celeb-1M for only classification.

Evaluation: In all the three datasets, we randomly split them into 80% training samples and 20% testing samples. In classification, we evaluate the top1 accuracy based on if the maximum class prediction matches the correct class label[23], and the results on the LFW[24] database (6000 pairs) are also reported. In alignment, the Normalized Root Mean Squared Error (NRMSE) is widely used to evaluate the alignment performance[42]; while in verification, we compute the Euclidean distance between each pair in testing samples, and the top1 accuracy is reported base on if a test sample and its nearest sample belong to the same class.

Teacher and Student: To learn the large number of identities, we use ResNet-50[16] as the teacher network, which is deep enough to handle our problem. For the student networks, given the fact that deeper students are better for model distillation[1, 41, 33], we remain the same depth to the teacher network, while only reducing the number of parameters for each layer. Consider that different datasets have different number of identities and different tasks have different difficulties, we adopt three student networks by dividing the number of convolution kernels in each layer by 2, 4 and 8, which give ResNet-50/2, ResNet-50/4 and ResNet-50/8 respectively.

Pre-processing and Training: Given an image, we first resize it to 256×256, then a sub-image with 224×224 is randomly cropped and flipped, and 224×224 is used for all the tasks. Particularly, we use no mean subtraction or image whitening, as we put a batch normalization layer right after the input data to learn the normalization parameters. In training, the batchsize is set to be 256, 64 and 128 for classification, alignment and verification respectively, and the Nesterov Accelerated Gradient(NAG) is adopted for optimization, which is found to converge much quickly than SGD. For the learning rate, if the network is trained from scratch, 0.1 is used at the beginning; while if the network is initialized, 0.01 is used to continue, and networks are trained by 30 epochs in each learning rate to fully converge. Besides, when training student networks, we adopt the online learning between teacher and student networks, *i.e.,* the output of the teacher network is generated online. Finally, the balancing weight λ and temperature τ are set by cross-validation.

Symbols in Experiments: We define some symbols for clarity in experiments. In the classification task, we denote $Scratch$ to be the initialized network $\mathbf{W}_{\mathbf{S_0}}^{\mathbf{cls}}$, and $Distill$ to be the distilled network $\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}$. For the non-classification tasks, $Scratch$ is the model trained by its own true labels, while $Pretrain$ and $Distill$ take $\mathbf{W}_{\mathbf{S_0}}^{\mathbf{cls}}$ and $\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}$ as initialization for alignment and verification. Particularly, only the models trained in CASIA-WebFace is used for initialization. Besides, $Hidden$, $Soft$ and $Label$ represent the top hidden layer $\mathbf{H}_{\mathbf{T}}$, soft predictions $\mathbf{P}_{\mathbf{T}}^{\tau}$ and class labels $\mathbf{y}^{\mathbf{cls}}$ respectively, and /(2,4,8) is the abbreviation for ResNet-50/(2,4,8).

4.2 Face classification

Table.1 shows the distillation results using different teacher supervision and student initialization. It can be observed that among different supervision, soft predictions obtain the highest accuracy that is 1% higher than using the top hidden layer and logits; while among different initialization, the full layer initialization achieves the accuracy of 75.06%, which improves a large margin over others. These results demonstrates the superiority of soft predictions and full layer initialization.

Table 1: The top1 accuracy of model distillation in classification. Results are obtained on CASIA-WebFace. Left: different teacher supervision; Right: different initialization.

Scratch/8	+ Hidden[27]	+ Logits[1]	+ Soft[18]	Distill/8	+ Scratch	+ Fit-Half[33]	+ Full
Top1(%)	60.08	59.77	61.27	Top1(%)	64.49	69.88	75.06

Base on this evidence, Table.2 shows classification results of model distillation on CASIA-WebFace and MS-Celeb-1M, and we have two main observations. Firstly, the model distillation can always obtain large improvements over the initialized model, *e.g.,* $Distill/8$ is more than 9% higher than $Scratch/8$ in top1 accuracy on both datasets. Secondly, on CASIA-WebFace, $Distill/4$ is competitive and $Distill/2$ is higher than the teacher by a large margin; while on MS-Celeb-1M, $Distill/2$ is competitive to the teacher. It indicates that more identities need more parameters to learn, and we can select the appropriate compression rates for our own systems. These results demonstrate the proposed method can improve the student model to a much higher level which can be competitive to the teacher, and even surpass the teacher under appropriate compression rates.

Table 2: The top1 and LFW accuracy of model distillation in the classification task. Results are obtained on CASIA-WebFace (left) and MS-Celeb-1M (right).

CASIA-WebFace	Teacher	Scratch/2	Scratch/4	Scratch/8	***MS-Celeb-1M***	Teacher	Scratch/2	Scratch/4	Scratch/8
Top1(%)	88.61	82.25	79.36	66.12	Top1(%)	90.53	84.59	81.94	57.84
		Distill/2	Distill/4	Distill/8			Distill/2	Distill/4	Distill/8
Top1(%)		**91.01**	87.21	75.06	Top1(%)		88.38	85.26	70.98
	Teacher	Scratch/2	Scratch/4	Scratch/8		Teacher	Scratch/2	Scratch/4	Scratch/8
LFW(%)	97.67	97.27	96.7	95.12	LFW(%)	99.11	98.61	98.03	96.33
		Distill/2	Distill/4	Distill/8			Distill/2	Distill/4	Distill/8
LFW(%)		**98.2**	**97.57**	96.18	LFW(%)		98.88	98.18	96.98

4.3 Face alignment

Table.3 shows the alignment results of ResNet-50/8 with different initialization and supervision on the CelebA dataset. The reason we only consider ResNet-50/8 is that alignment is a relatively simple problem and most studies use shallow and small networks, thus a large compression rate is necessary for ResNet-50. Particularly, only the classification models trained in CASIA-WebFace are used for initialization. It can be observed that the $Pretrain$ and $Distill$ models obtain much lower error rates, *e.g.,* they are 2% lower than $Scratch/8$ in NRMSE. Besides, we further see that in $Pretrain/8$ and $Distill/8$, adding the soft predictions as supervision will harm the performance, while the top hidden layer can always be beneficial. This result validates our analysis in Sec.3.3.1 that the class identity in soft predictions cannot give useful information as there is large difference in race and region between the CASIA-WebFace and CelebA datasets. Finally, $Distill/8 + Hidden$ gives the best performance with the NRMSE of 3.21%, which is competitive to the teacher. This is encouraging because this competitive result is obtained under the large compression rate, which makes it potential to use smaller networks in practical applications.

Table 3: The NRMSE of model distillation in alignment. Results are obtained on CelebA.

CelebA	Scratch/8	+ Hidden	+ Soft	Pretrain/8	+ Hidden	+ Soft	Distill/8	+ Hidden	+ Soft	Teacher
NRMSE(%)	5.41	3.53	4.05	3.36	3.24	3.60	3.29	3.21	3.54	3.02

4.4 Face verification

Table.4 shows the verification results of different initialization and supervision on the CASIA-WebFace dataset. Particularly, the triplet models in Table.4 are trained with the online hard negative mining as in Facenet[35], and only the classification models trained in CASIA-WebFace are used for initialization. Besides, the LFW accuracy is not reported because it uses only 6000 pairs of images which may bring some randomness, while we use almost 100000 test images in CASIA-WebFace, thus the small difference in top1 accuracy may not be correctly shown in LFW. Base on Table.4, we have three main observations. Firstly, models trained with class labels are better, *i.e.,* $Pretain + Label$ and $Distill + Label$ are always better than $Pretrain$ and $Distill$, which implies that class labels can retain the class identity as the verification and classification have the same class labels. Secondly, it can be clearly observed that adding the top hidden layer as supervision will harm the performance, while the soft predictions are always beneficial, and this is opposite to the results in alignment. As analyzed in Sec.3.3.2, soft predictions are beneficial due to the same class labels to classification, while the top hidden layer will be easily over-fitting with hard negative mining. Table.5 gives the results with random negative sampling. Although small improvements are obtained with the top hidden layer, the top1 accuracy is much lower than the one with negative mining, thus we prefer to use the online hard negative mining in training. Finally, we observe that $Distill/2 + Label + Soft$ has surpassed the teacher by a large margin, which demonstrates the effectiveness of the proposed method in model distillation on the verification task.

Table 4: The top1 accuracy of model distillation with online hard negative mining in the verification task. Results are obtained on CASIA-WebFace.

CASIA Top1(%)	Pretrain/2 63.98	Pretrain/4 61.74	Pretrain/8 51.03	Pretrain/2 + Label 72.38	Pretrain/4 + Label 66.64	Pretrain/8 + Label 51.86
Top1(%)	+ Hidden 60.66	+ Hidden 61.71	+ Hidden 49.19	+ Hidden 70.54	+ Hidden 65.08	+ Hidden 51.43
Top1(%)	+ Soft 66.5	+ Soft 62.64	+ Soft 51.76	+ Soft 73.62	+ Soft 68.24	+ Soft 53.45
Top1(%)	Distill/2 71.29	Distill/4 68.17	Distill/8 56.69	Distill/2 + Label **79.51**	Distill/4 + Label 72.01	Distill/8 + Label 57.66
Top1(%)	+ Hidden 68.74	+ Hidden 66.74	+ Hidden 53.99	+ Hidden **77.63**	+ Hidden 70.31	+ Hidden 56.87
Top1(%)	+ Soft 71.23	+ Soft 68.12	+ Soft 56.52	+ Soft **79.96**	+ Soft 72.82	+ Soft 57.78
Top1(%)	Teacher 74.16	Teacher 74.15	Teacher 73.81	Teacher 74.16	Teacher 74.15	Teacher 73.81

Table 5: The top1 accuracy of model distillation with random negative sampling in the verification task. Results are obtained on CASIA-WebFace.

CASIA	Random/8	Random/8 + Hidden	Random/8 + Label	Random/8 + Label + Hidden
Top1(%)	40.16	40.95	43.83	43.92

5 Discussion

We have only focused on the face recognition problem in this paper, while the proposed method can also be applied in more general tasks. In our case, face alignment is actually a regression problem with Euclidean loss, and face verification is a metric learning task with triplet loss, thus general tasks with Euclidean loss and triplet loss can also fit in our framework, *e.g.,* image classification in ImageNet[8], object detection with regression targets, and image retrieval with triplet loss. Usually, most these domain-similar tasks can use pretrain classification models on ImageNet[32, 47]. Therefore, similar to face recognition, a good distillation model on ImageNet is the key factor for knowledge distillation in object detection and retrieval.

Table.6 gives the results of model distillation on ImageNet, and the results on the SUN-397 database[43] are also reported to test the generality of our method on scene classification. Particularly, the top1 accuracy on ImageNet is reported based on its validation set; while on SUN-397, we adopt a similar way as in face classification that 80% for training and 20% for testing. It can be observed that all the $Distill$ models have obtained some improvements over the $Scratch$ models, and the student can be competitive to the teacher or surpasses the teacher under the compression rate of 2, *e.g.,* $Distill/2$ is approaching teacher on ImageNet, while it exceeds the teacher on SUN-397. However, we also see that the improvement of $Distill$ over $Scratch$ is very small, which is far less than the improvement in face classification in Table.2.

There are two main reasons that may be responsible. Firstly, the intra-class difference is large. In face classification, most images are near-frontal faces with some variances in viewpoint, illumination and expression; while in ImageNet, there are more than 1000 images for each class, and objects vary a lot in scale, shape, color, viewpoint, occlusion and illumination. As a result, distilling the knowledge is more difficult. Secondly, backgrounds may have negative influence. In face classification, images are cropped and aligned to guarantee the clean faces for training; while in ImageNet and SUN-397, we learn not only the class identities, but also their background information, thus this will also make the student model learn these backgrounds, which may be confused. In the future work, we will consider how to obtain good knowledge distillation in more general tasks.

Table 6: The top1 accuracy of model distillation in the classification task. Results are obtained on ImageNet[8] and SUN-397[43].

ImageNet	Teacher	Scratch/2	Scratch/4	Scratch/8	***SUN-397***	Teacher	Scratch/2	Scratch/4	Scratch/8
Top1(%)	72.95	69.18	63.58	51.01	Top1(%)	61.58	60.48	57.35	55.1
		Distill/2	Distill/4	Distill/8			Distill/2	Distill/4	Distill/8
Top1(%)		70.82	64.41	51.74	Top1(%)		61.87	59.61	57.14

6 Conclusion

In this paper, we focus on the face recognition problem, and propose the model distillation with full layer student initialization and knowledge transfer. The distillation method can not only be used in face classification, but also in some domain-similar tasks including face alignment and verification, which have never been considered in model distillation. Firstly, in face classification, we initialize all layers of the student network with class labels, then these class labels and additional soft predictions from teacher network are used jointly to train a better student. Secondly, to transfer the distilled knowledge to face alignment and verification, we take the teacher and student models as initialization for training new teachers and students in alignment and verification, which use the top hidden layer and soft predictions as teacher supervision respectively. Experiments on the CASIA-WebFace, CelebA and large-scale MS-Celeb-1M datasets have demonstrated the effectiveness of the proposed method, which can train student networks that be competitive or surpass the teacher under appropriate compression rates. Moreover, we discuss the generality of the proposed method in more general tasks, which needs to be further studied in our future work.

References

[1] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. 2006.

[3] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

[4] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. In *arXiv:1605.07678*, 2017.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[6] KyungHyun Cho, Tapani Raiko, Alexander Ilin, and Juha Karhunen. A two-stage pretraining algorithm for deep boltzmann machines. 2012.

[7] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. In *arXiv:1602.02830*, 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] Emily Denton, Wojciech Zaremba, Yann LeCun, and Yann LeCun. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.

[10] Michael Figurnov, Maxwell D. Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *arXiv:1612.02297*, 2016.

[11] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization. In *ICLR*, 2015.

[12] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[13] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

[14] Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Enhao Gong, Shijian Tang, Erich Elsen, Peter Vajda, Manohar Paluri, John Tran, Bryan Catanzaro, and William J. Dally. Dsd: Dense-sparse-dense training for deep neural networks. In *ICLR*, 2017.

[15] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *NIPS*, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] G. E. Hinton and R. R. Salaktudinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Workshop*, 2014.

[19] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006.

[20] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. In *arXiv:1602.07360*, 2016.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[22] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. In *Technical report, University of Toronto*, 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[24] Erik Learned-Miller, Gary B. Huang, Aruni RoyChowdhury, and Gang Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, 2016.

[25] Lanlan Liu and Jia Deng. Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution. In *arXiv:1701.00299*, 2017.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[27] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. Face model compression by distilling knowledge from neurons. In *AAAI*, 2016.

[28] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *NIPS*, 2015.

[29] George Papamakarios. Distilling model knowledge. In *arXiv:1510.02437*, 2015.

[30] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.

[31] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[33] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[34] Bharat Bhusan Sau and Vineeth N. Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. In *arXiv:1610.09650*, 2017.

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[39] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, and Weinan E. Convolutional neural networks with low-rank regularization. In *CoPR*, 2015.

[40] Surat Teerapittayanon and Bradley McDanel. Branchynet: Fast inference via early exiting from deep neural networks. In *ICPR*, 2016.

[41] Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? In *arXiv:1603.05691*, 2017.

[42] Yue Wu, Tal Hassner, KangGeon Kim, Gerard Medioni, and Prem Natarajan. Facial landmark detection with tweaked convolutional neural networks. In *arXiv:1511.04031*, 2015.

[43] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[44] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. In *arXiv:1506.02640*, 2016.

[45] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? 2014.

[46] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection. *TPAMI*, 2016.

[47] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, 2015.