Model Distillation with Knowledge Transfer from Face Classification to Alignment and Verification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Knowledge distillation is a potential solution for model compression. The idea is to make a small student network imitate the target of a large teacher network, then the student network can be competitive to the teacher one. Most previous studies focus on model distillation in the classification task, where they propose different architects and initializations for the student network. However, only the classification task is not enough, and other related tasks such as regression and retrieval are barely considered. To solve the problem, in this paper, we take face recognition as a breaking point and propose model distillation with knowledge transfer from face classification to alignment and verification. By selecting appropriate initializations and targets in the knowledge transfer, the distillation can be easier in non-classification tasks. Experiments on the CelebA and CASIA-WebFace datasets demonstrate that the student network can be competitive to the teacher one in alignment and verification, and even surpasses the teacher network under specific compression rates. In addition, to achieve stronger knowledge transfer, we also use a common initialization trick to improve the distillation performance of classification. Evaluations on the CASIA-Webface and large-scale MS-Celeb-1M datasets show the effectiveness of this simple trick.

1 Introduction

Since the emergence of Alexnet[19], larger and deeper networks have shown to be more powerful[31, 32, 13]. However, as the network going larger and deeper, it becomes difficult to use it in 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[24, 3, 1, 35, 22, 29, 14, 28], weight quantization[8, 6, 26], weight pruning[12, 10, 11, 16, 33] and weight decomposition[40, 4, 7, 34, 23]. 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 network and a small student one, and the objective is to make the student network competitive to the teacher one by learning the specific target of the teacher network. Previous studies mainly consider the selection of targets in the classification task, *e.g.,* hidden layers[22], logits[1, 35, 29] or soft predictions[14, 28]. However, only the distillation of the classification task is not enough, and some common tasks such as regression and retrieval should also be considered. In this paper, we take face recognition as a breaking point that we start with the knowledge distillation in face classification, and consider the distillation on two domain-similar tasks, including face alignment and verification. The objective of face alignment is to locate the key-point locations in each image; while in face verification, we have to determine if two images belong to the same identity.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

For distillation on non-classification tasks, one intuitive idea is to adopt a similar method as in face classification that trains teacher and student networks from scratch. In this way, the distillation on all these tasks will be independent, and this is a possible solution. However, this independence cannot give the best distillation performance. There has been strong evidence that in object detection[27], object segmentation[5] and image retrieval[41], they all used the pretrained classification model(on ImageNet) as initialization to boost performance. This success comes from the fact that their domains are similar, which makes them transfer a lot from low-level to high-level representation[39]. Similarly, face classification, alignment and verification also share the similar domain, thus we transfer the distilled knowledge of classification by taking its teacher and student networks to initialize corresponding networks in alignment and verification.

Another problem in knowledge transfer is what targets should be used for distillation? In face classification, the knowledge is distilled from the teacher network by learning its soft-prediction, which has been proved to work well[14, 28]. However, in face alignment[37] and verification[37], they have additional task-specific targets for learning. As a result, selecting the classification or task-specific target for distillation remains a problem. One intuitive idea is to measure the relevance of objectives between non-classification and classification tasks. For example, it is not obvious to see the relation between face classification and alignment, but the classification can help a lot in verification. Therefore, it seems reasonable that if the tasks are highly related, the classification target is preferred, or the task-specific target is better.

Inspired by the above thoughts, in this paper, we propose the model distillation in face alignment and verification by transferring the distilled knowledge from face classification. With appropriate selection of initializations and targets, we show that the distillation performance of alignment and verification on the CelebA[21] and CASIA-WebFace[38] datasets can be largely improved, and the student network can even exceed the teacher network under specific compression rates.

This knowledge transfer is our main contribution. In addition, we realize that in the proposed method, the knowledge transfer depends on the distillation of classification, thus we use a common initialization trick to further boost the distillation performance of classification at the beginning. Evaluations on the CASIA-WebFace[38] and large-scale MS-Celeb-1M[9] datasets show that this simple trick can give the best distillation results in the classification task.

2 Related Work

In this part, we introduce some previous studies on knowledge distillation. Particularly, all the following studies focus on the task of classification. Buciluǎ *et al.*[3] propose to generate synthetic data by a teacher network, then a student network is trained with the synthetic data to mimic the identity labels of the teacher network. However, Ba and Caruana[1] observe that these labels have lost the uncertainties of the teacher network, thus they propose to regress the logits (pre-softmax activations)[14]. Besides, they prefer the student network to be deep, which is good to mimic complex functions. To better learn the function, Gregor *et al.*[35] observe the student network should not only be deep, but also be convolutional, and they get competitive performance to the teacher network in CIFAR[18]. Most methods need a large ensemble of teacher networks for distillation, but this will take a long training and inference time[29]. To address the issue, Sau and Balasubramanian[29] propose noise-based regularization that can simulate the logits of multiple teacher networks. However, Luo *et al.*[22] observe the values of the logits are unconstrained, and the high dimensionality will also cause fitting problem. As a result, they use the hidden layer as it captures as much information as the logits but is more compact.

All these methods only use the targets of the teacher network in distillation, while if the target is not confident, the training of the student network will be difficult. To solve the problem, Hinton *et al.*[14] propose a multi-task approach which uses identity labels and the target of the teacher network jointly. Particularly, they use the post-softmax activations with temperature smoothing as the target, which can better represent the label distribution. One problem is that student networks are mostly trained from scratch. Given the fact that initialization is important, Romero *et al.*[28] propose to initialize the shallow layers of the student network by regressing the mid-level target of the teacher network. However, these studies only consider the knowledge distillation in classification, which largely limits its application in model compression. In this paper, we consider the face recognition problem as a breaking point and extend the distillation to non-classification tasks.

3 Distillation of Classification

Due to the proposed knowledge transfer depends on the distillation in classification, improving the classification itself is necessary. In this part, we first review the idea of distillation for classification, then introduce how to boost it by a simple initialization trick.

3.1 Review of Knowledge Distillation

We adopt the distillation framework in Hinton *et al.*[14], 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, 35, 29]. However, the post-softmax predictions have lost some relative uncertainties that are more informative, thus a temperature parameter τ is used to smooth predictions $\mathbf{P_T}$ and $\mathbf{P_S}$ to be $\mathbf{P}_\mathbf{T}^\tau$ and $\mathbf{P}_\mathbf{S}^\tau$, which are denoted as ***soft predictions***:

$$\mathbf{P}_\mathbf{T}^\tau = \mathrm{softmax}(\mathbf{a_T}/\tau), \quad \mathbf{P}_\mathbf{S}^\tau = \mathrm{softmax}(\mathbf{a_S}/\tau). \tag{1}$$

Then, consider $\mathbf{P}_\mathbf{T}^\tau$ as the target, knowledge distillation optimizes the following loss function

$$\mathrm{L}(\mathbf{W_S^{cls}}) = \mathrm{H}(\mathbf{P_S}, \mathbf{y^{cls}}) + \alpha \mathrm{H}(\mathbf{P}_\mathbf{S}^\tau, \mathbf{P}_\mathbf{T}^\tau), \tag{2}$$

wherein $\mathbf{W_S^{cls}}$ is the parameter of the student network, and $\mathbf{y^{cls}}$ is the identity label. For simplicity, we omit min and the number of samples N, and denote the upper right symbol cls as the classification task. In addition, $\mathrm{H}(,)$ is the cross-entropy, thus the first term is the softmax loss, while the second one is the cross-entropy between the soft predictions of the teacher and student network, with α balancing between the two terms. This multi-task training is advantageous because the target $\mathbf{P}_\mathbf{T}^\tau$ cannot be guaranteed to be always correct, and if the target is not confident, the identity label $\mathbf{y^{cls}}$ will take over the training of the student work.

3.2 Initialization Trick

It is noticed that in Eqn.(2), the student network is trained from scratch. As demonstrated in [1, 35] that deeper student networks are better for distillation, initialization thus has become very important[15, 2, 17]. Based on the evidence, Fitnet[28] first initializes the shallow layers of the student network by regressing the mid-level target of the teacher network, then it follows Eqn.(2) for distillation. However, only initializing the shallow layers is still difficult to learn high-level representation, which is generated by deep layers. Furthermore, [39] shows that the network transferability increases as tasks become more similar. In our case, the initialization and distillation are both classification tasks with exactly the same data and identity labels, thus more deep layers should be initialized for higher transferability, and we use a simple trick to achieve this.

To obtain an initial student network, we first train it with softmax loss:

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

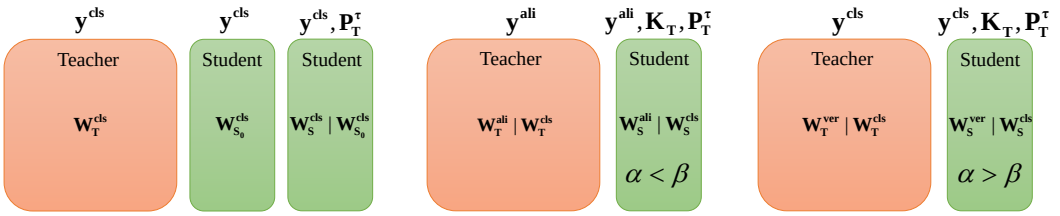
wherein the lower right symbol S_0 denotes the initialization for student network S. In this way, the student network is fully initialized. Then, we modify Eqn.(2) as

$$\mathrm{L}(\mathbf{W_S^{cls}}|\mathbf{W_{S_0}^{cls}}) = \mathrm{H}(\mathbf{P_S}, \mathbf{y^{cls}}) + \alpha \mathrm{H}(\mathbf{P}_\mathbf{S}^\tau, \mathbf{P}_\mathbf{T}^\tau), \tag{4}$$

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. This process is shown in Fig.1(a). It can be seen that the only difference with Eqn.(2) is that the student network is trained with the full initialization, and this simple trick has been commonly used, *e.g.,* initializing the VGG-16 model[31]. We later show that this trick can get promising improvements over Eqn.(2) and Fitnet[28].

4 Distillation Transfer

In this part, we show how to transfer the distilled knowledge from face classification to face alignment and verification. The knowledge transfer consists of two steps: transfer initialization and target selection, which are elaborated as follows.



(a) Classification (b) Alignment (c) Verification

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

4.1 Transfer Initialization

The first step of transfer is the initialization. The motivation is based on the evidence that in detection, segmentation and retrieval, they have used the pretrained classification model (on ImageNet) as initialization to boost performance[27, 5, 41]. The availability of this idea comes from the fact that they share the similar domain, which makes them transfer easily from low-level to high-level representation[39]. Similarly, the domains of face classification, alignment and verification are also similar, thus we can transfer the distilled knowledge in the same way.

For simplicity, we denote the parameters of teacher and student networks in face classification as $\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}$ and $\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}$. Analogically, they are $\mathbf{W}_{\mathbf{T}}^{\mathbf{ali}}$ and $\mathbf{W}_{\mathbf{S}}^{\mathbf{ali}}$ in alignment, while $\mathbf{W}_{\mathbf{T}}^{\mathbf{ver}}$ and $\mathbf{W}_{\mathbf{S}}^{\mathbf{ver}}$ in verification. As analyzed above, in distillation of alignment and verification, teacher and student networks will be initialized by $\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}$ and $\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}$ respectively.

4.2 Target Selection

Based on the initialization, the second step is to select appropriate targets in the teacher network for distillation. One problem is that non-classification tasks have their own task-specific targets, but given the additional soft predictions $\mathbf{P}_{\mathbf{T}}^{\tau}$, which one should we use? To be clear, we first propose the general distillation for non-classification tasks as follows:

$$\mathrm{L}(\mathbf{W}_{\mathbf{S}}|\mathbf{W}_{\mathbf{S}}^{\mathbf{cls}}) = \Phi\left(\mathbf{W}_{\mathbf{S}}, \mathbf{y}\right) + \alpha \mathrm{H}(\mathbf{P}_{\mathbf{S}}^{\tau}, \mathbf{P}_{\mathbf{T}}^{\tau}) + \beta \Psi\left(\mathbf{K}_{\mathbf{S}}, \mathbf{K}_{\mathbf{T}}\right), \tag{5}$$

where $\mathbf{W}_{\mathbf{S}}$ and $\mathbf{y}$ denote the task-specific network parameter and label respectively. $\Phi\left(\mathbf{W}_{\mathbf{S}}, \mathbf{y}\right)$ is the task-specific loss function, and $\Psi\left(\mathbf{K}_{\mathbf{S}}, \mathbf{K}_{\mathbf{T}}\right)$ is the task-specific distillation term with the targets selected as $\mathbf{K}_{\mathbf{T}}$ and $\mathbf{K}_{\mathbf{S}}$ in teacher and student networks. Besides, α and β are the balancing terms between classification and non-classification tasks. In Eqn.(5), the above problem has become how to set α and β for a given non-classification task. In the following two parts, we will give some discussions on two tasks: face alignment and verification.

4.2.1 Alignment

The task of face alignment is to locate the key-point locations for each image. Without loss of generality, there is no any identity labels, but only the keypoint locations for each image. Face alignment is usually considered as a regression problem[37], thus we train the teacher network with optimizing the Euclidean loss:

$$\mathrm{L}(\mathbf{W}_{\mathbf{T}}^{\mathbf{ali}}|\mathbf{W}_{\mathbf{T}}^{\mathbf{cls}}) = \left\|\mathbf{R}_{\mathbf{T}} - \mathbf{y}^{\mathbf{ali}}\right\|^2, \tag{6}$$

wherein $\mathbf{R}_{\mathbf{T}}$ is the regression prediction of the teacher network and $\mathbf{y}^{\mathbf{ali}}$ is the regression label. In distillation, except for the available soft predictions $\mathbf{P}_{\mathbf{T}}^{\tau}$(classification target), another one is the task-specific target that can be the hidden layer $\mathbf{K}_{\mathbf{T}}$[22], and it satisfies $\mathbf{R}_{\mathbf{T}} = fc\left(\mathbf{K}_{\mathbf{T}}\right)$ with fc being a fully-connected mapping.

In face classification, the key in distinguishing different identities is the appearance around the key-points such as shape and color, but the difference of key-point locations for different identities is tiny. As a result, face identity is not the main influencing factor for these locations, but it is still related as different identities may have slightly different locations. Instead, pose and viewpoint

variations have a much larger influence. Therefore, in face alignment, the hidden layer is preferred for distillation, which gives Eqn.(7) by setting $\alpha < \beta$, as shown in Fig.1(b).

$$\mathrm{L}(\mathbf{W_S^{ali}}|\mathbf{W_S^{cls}}) = \left\|\mathbf{R_S} - \mathbf{y^{ali}}\right\|^2 + \alpha \mathrm{H}(\mathbf{P_S^\tau}, \mathbf{P_T^\tau}) + \beta \|\mathbf{K_S} - \mathbf{K_T}\|^2. \tag{7}$$

4.2.2 Verification

The task of face verification is to determine if two images belong to the same identity. In verification, triplet loss[25, 30] is a widely used metric learning method[30], and we take it for model distillation. Without loss of generality, we have the same identity labels as in classification[38, 9], then the teacher network can be trained as

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

where $\mathbf{K_T^a}$, $\mathbf{K_T^p}$ and $\mathbf{K_T^n}$ are the hidden layers for the anchor, positive and negative samples respectively, *i.e.,* a and p have the same identity, while a and n come from different identities. Besides, λ controls the margin between positive and negative pairs.

Similar to face alignment, we consider the hidden layer $\mathbf{K_T}$ and soft prediction $\mathbf{P_T^\tau}$ as two possible targets in distillation. In fact, classification focuses on the difference of identities, *i.e.* the inter-class relation, and this relation can easily tell if two image have the same identity. As a result, classification can be beneficial to boost the performance of verification. Therefore, in face verification, the soft predictions are preferred for distillation, which gives the following loss function by setting $\alpha > \beta$, as shown in Fig.1(c).

$$\mathrm{L}(\mathbf{W_S^{ver}}|\mathbf{W_S^{cls}}) = \left[\|\mathbf{K_S^a} - \mathbf{K_S^p}\|^2 - \|\mathbf{K_S^a} - \mathbf{K_S^n}\|^2 + \lambda\right]_+ + \alpha \mathrm{H}(\mathbf{P_S^\tau}, \mathbf{P_T^\tau}) + \beta \|\mathbf{K_S} - \mathbf{K_T}\|^2. \tag{9}$$

Particularly, some studies[36] also shows the benefits by using softmax loss in Eqn.(8). For comparison, we also add the softmax loss $\mathrm{H}(\mathbf{P_T}, \mathbf{y^{cls}})$ and $\mathrm{H}(\mathbf{P_S}, \mathbf{y^{cls}})$ in Eqn.(8) and Eqn.(9) respectively for further enhancement.

4.2.3 A Short Summary

As analyzed above, α and β should be set differently in the distillation of different tasks. The key is to measure the relevance of objectives between classification and non-classification tasks. For a given task, if the classification is highly related, then $\alpha > \beta$ is necessary, or $\alpha < \beta$ should be set. Though this rule cannot be theoretically guaranteed, it provides some guidelines to use knowledge distillation in more non-classification tasks.

5 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 knowledge distillation in the tasks of face classification, alignment and verification.

5.1 Experimental Setup

Database: We use three popular datasets for evaluation, including CASIA-WebFace[38], CelebA[21] and MS-Celeb-1M[9]. CASIA-WebFace contains 10575 people and 494414 images, while CelebA has 10177 people with 202599 images and the label of 5 key-point locations. Compared to the previous two, MS-Celeb-1M is a large-scale dataset that contains $100K$ people with 8.4 million images. In experiments, we use CASIA-WebFace and MS-Celeb-1M for classification, CelebA for alignment and CASIA-WebFace for verification.

Evaluation: In all datasets, we randomly split them into 80% training and 20% testing samples. In classification, we evaluate the top1 accuracy based on if the identity of the maximum prediction matches the correct identity label[19], and the results on the LFW[20] database (6000 pairs) are also reported by computing the percentage of how many pairs are correctly verified. In alignment, the Normalized Root Mean Squared Error (NRMSE) is widely used to evaluate the alignment performance[37]; while in verification, we compute the Euclidean distance between each pair in

testing samples, and the top1 accuracy is reported based on if a test sample and its nearest sample belong to the same identity. Particularly, LFW is not used in verification because 6000 pairs are not enough to see the difference obviously for different methods.

Teacher and Student: To learn the large number of identities, we use ResNet-50[13] as the teacher network, which is deep enough to handle our problem. For student networks, given the fact that deep student networks are better for knowledge distillation[1, 35, 28], we remain the same depth but divide 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 resize it to 256×256 wherein a sub-image with 224×224 is randomly cropped and flipped. Particularly, we use no mean subtraction or image whitening, as we use batch normalization right after the input data. 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 faster convergence. For the learning rate, if the network is trained from scratch, 0.1 is used; while if the network is initialized, 0.01 is used to continue, and 30 epochs are used in each rate. Besides, in distillation, student networks are trained with the targets of the teacher network generated online, and the temperature τ and margin λ are set to be 3 and 0.4 by cross-validation. Finally, the balancing terms α and β have many possible combinations, and we show later how to avoid this by an experimental trick.

Symbols in Experiments: (1)$Scratch$: student networks are not initialized; (2)$Pretrain$: student networks are trained with task-specific initialization; (3)$Distill$: student networks are initialized with $\mathbf{W_S^{ls}}$; (4)$Soft$: the soft prediction $\mathbf{P_T^{\tau}}$; (5)$Hidden$: the hidden layer $\mathbf{K_T}$.

5.2 Comparison to Previous Studies

In this part, we compare the initialization trick to previous studies in classification. Table.1 shows the comparison of different targets and initializations. It can be observed from the first table that without any initialization, soft predictions achieve the best distillation performance, *i.e.*, 61.27%. Based on the best target, the second table gives the results of different initializations in distillation. We see that our full initialization obtains the best accuracy of 75.06%, which is much higher than other methods, *i.e.*, 10% and 5% higher than the $Scratch$ and Fitnet[28]. This shows that the full initialization of student networks can give the highest transferability in classification, and also demonstrates the effectiveness of this simple trick.

Table 1: The comparison to previous studies with different initializations and targets. Results are given on CASIA-WebFace.

CASIA-WebFace	Teacher Network	Student Network with Different Targets				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
	88.61	ResNet-50/8	Scratch	Hidden[22] 60.08	Logits[29] 59.77	Soft[14] ***61.27***

CASIA-WebFace	Teacher Network	Student Network with Different Initializations				
Top1 acc(%)	ResNet-50	Network	Learning	Initialization		
	88.61	ResNet-50/8	Distill	Scratch[14] 64.49	Fitnet[28] 69.88	Ours ***75.06***

5.3 Face Classification

Base on the best initialization and target, Table.2 shows the distillation results of face classification on CASIA-WebFace and MS-Celeb-1M, and we have three main observations. Firstly, the student networks trained with full initialization can obtain large improvements over the ones trained from scratch, which further demonstrates the effectiveness of the initialization trick in large scale cases. Secondly, some student networks can be competitive to the teacher network or even exceed the teacher one by a large margin, *e.g.,* in CASIA-WebFace, ResNet-50/4 can be competitive to the teacher network, while ResNet-50/2 is about 3% higher than the teacher one in both top1 and LFW accuracy. Finally, in the large-scale MS-Celeb-1M, student networks cannot exceed the teacher network but

only be competitive, which shows that the knowledge distillation is still challenging in distilling a large number of identities.

Table 2: The top1 and LFW accuracy of knowledge distillation in classification. Results are obtained on CASIA-WebFace and MS-Celeb-1M.

CASIA-WebFace	Teacher Network	Student Network			
	ResNet-50	Initialization	ResNet-50/2	ResNet-50/4	ResNet-50/8
Top1 acc(%)	88.61	Scratch[14]	82.25	79.36	66.12
		Ours	***91.01***	87.21	75.06
LFW acc(%)	97.67	Scratch[14]	97.27	96.7	95.12
		Ours	***98.2***	97.57	96.18
MS-Celeb-1M	Teacher Network	Student Network			
	ResNet-50	Initialization	ResNet-50/2	ResNet-50/4	ResNet-50/8
Top1 acc(%)	90.53	Scratch[14]	84.59	81.94	57.84
		Ours	***88.38***	85.26	70.98
LFW acc(%)	99.11	Scratch[14]	98.61	98.03	96.33
		Ours	***98.88***	98.18	96.98

5.4 Face Alignment

In this part, we give the evaluation of distillation in face alignment. Table.3 shows the distillation results of ResNet-50/8 with different initializations and targets on CelebA. The reason we only consider ResNet-50/8 is that face alignment is a relatively easy problem and most studies use shallow and small networks, thus a large compression rate is necessary for the deep ResNet-50. One important thing is how to set α and β in Eqn.(7). As there are many possible combinations, we use a simple trick by measuring their individual influence and discard the target with the negative impact by setting $\alpha = 0$ or $\beta = 0$; while if they both have positive impacts, $\alpha > 0, \beta > 0$ should be set to keep both targets in distillation.

As shown in Table.3, when the initializations of $Pretrain$ and $Distill$ are used, $\alpha = 1, \beta = 0$(soft prediction) always decreases performance, while $\alpha = 0, \beta = 1$(hidden layer) gets consistent improvements, which implies that the hidden layer($\alpha = 0, \beta = 1$) is preferred in the distillation of face alignment. Therefore, $\alpha = 0, \beta = 1$ is used in face alignment. It can be observed in Table.3 that $Distill$ has a lower error rate than $Pretrain$, which shows that $\mathbf{W_S^{cls}}$ has higher transferability on high-level representation than the task-specific initialization. Besides, the highest distillation performance 3.21% is obtained by ResNet-50/8 with $Distill$ and $\alpha = 0, \beta = 1$, and it can be competitive to the one of the teacher network(3.02%).

Table 3: The NRMSE($\%$) of knowledge distillation in face alignment with different initializations and targets. Results are obtained on CelebA.

CelebA	Teacher Network	Student Network				
	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0, \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
NRMSE(%)	3.02	ResNet-50/8	Pretrain	3.36	***3.24***	3.60
			Distill	3.29	***3.21***	3.54

5.5 Face Verification

In this part, we give the evaluation of distillation in face verification. Similar to alignment, we select α and β in the same way. Table.4 shows the verification results of different initializations and targets on CASIA-WebFace, and the results are given by Eqn.(9). It can be observed that no matter which student network or initialization is used, $\alpha = 0, \beta = 1$(hidden layer) always decreases the baseline performance, while $\alpha = 1, \beta = 0$(soft prediction) remains almost the same. As a result, we discard the hidden layer and only use the soft prediction.

One interesting observation in Table.4 is that $\alpha = 0, \beta = 0$ always get the best performance, and the targets do not work at all. One possible reason is that the target in classification is not confident, *i.e.*, the top1 accuracy of ResNet-50 in classification is only 88.61%. To improve the classification ability, we add additional softmax loss in Eqn.(8) and Eqn.(9), and the results are shown in Table.5. We see that the accuracy of ResNet-50/2 and ResNet-50/4 has obtained remarkable improvements, which implies that the classification targets that are not confident cannot help the distillation. But with the additional softmax loss, the student work can adjust the learning by identity labels. As a result, $\alpha = 1, \beta = 0$ can get the best distillation performance, which is even much higher than the teacher network, *e.g.*, 79.96% of ResNet-50/2 with $Distill$ and $\alpha = 1, \beta = 0$.

Table 4: The top1 accuracy of distillation with single triplet loss in verification. Results are obtained on CASIA-WebFace.

CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	73.81	ResNet-50/2	Pretrain	63.98	60.66	66.50
			Distill	***71.29***	68.74	71.23
CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	73.81	ResNet-50/4	Pretrain	61.74	61.71	62.64
			Distill	***68.17***	66.74	68.12
CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	73.81	ResNet-50/8	Pretrain	51.03	49.19	51.76
			Distill	***56.69***	53.99	56.52

Table 5: The top1 accuracy of distillation with joint triplet loss and softmax loss in verification. Results are obtained on CASIA-WebFace.

CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	74.16	ResNet-50/2	Pretrain	72.38	70.54	73.62
			Distill	79.51	77.63	***79.96***
CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	74.16	ResNet-50/4	Pretrain	66.64	65.08	68.24
			Distill	72.01	70.31	***72.82***
CASIA-WebFace	Teacher Network	Student Network				
Top1 acc(%)	ResNet-50	Network	Initialization	Targets		
				$\alpha = 0,\ \beta = 0$	$\alpha = 0, \beta = 1$	$\alpha = 1, \beta = 0$
	74.16	ResNet-50/8	Pretrain	51.86	51.43	53.45
			Distill	57.66	56.87	***57.78***

6 Conclusion

In this paper, we take face recognition as a breaking point, and propose the knowledge distillation on two non-classification tasks, including face alignment and verification. We extend the previous distillation framework by transferring the distilled knowledge from face classification to face alignment and verification. By selecting appropriate initializations and targets, the distillation on non-classification tasks can be much easier. Besides, we also give some guidelines for target selection on non-classification tasks, and we hope these guidelines can be helpful for more tasks. Experiments on the CASIA-WebFace, CelebA and large-scale MS-Celeb-1M datasets have demonstrated the effectiveness of the proposed method, which gives the student networks that can be competitive or exceed the teacher network under appropriate compression rates. In addition, we use a common initialization trick to further improve the distillation performance of classification, and this can boost the distillation on non-classification tasks. Experiments on CASIA-WebFace have demonstrated the effectiveness of this simple trick.

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. In *NIPS*, 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] 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.

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

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

[9] 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.

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

[11] 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.

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

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

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

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

[16] 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.

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

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

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

[20] 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.

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

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

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

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

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

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

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

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

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

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

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

[32] 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.

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

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

[35] 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.

[36] Chong Wang, Xue Zhang, and Xipeng Lan. How to train triplet networks with 100k identities? In *arXiv:1709.02940*, 2017.

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

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

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

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

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