Deep Face Recognition in the Wild

Jing Yang

School of Computer Science
University of Nottingham

This dissertation is submitted for the degree of
Doctor of Philosophy

December 2021

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, **Dr. Georgios Tzimiropoulos** for his continuous guidance, support and encouragement throughout my PhD. I am grateful to Georgios for discussing the research topic, developing ideas, analysing the experiments, polishing the paper writing and keeping and bringing hope when I have lost mine. I sincerely thank him for his creativity, kindness, patience, and extensive knowledge.

Besides, I would like to thank my supervision team: Tony Pridmore and Michel Valstar who have assisted my studies in several circumstances. Tony is the leader of the Computer Vision Lab, and he has created an effective research environment for us. The weekly lab meetings provide researchers here with opportunities to share papers, discuss and develop research ideas. Michel is an excellent mentor who has given me valuable advice on the annual review, PhD planning and thesis schedule. Here, I would like to thank lab colleagues from B86: Aaron, Keerthy, Siyang, Kike, and Dimitris for their company and encouragement during my PhD.

Moreover, I would like to thank my coauthor: Adrian Bulat. I particularly thank him for spending time on numerous discussions on research ideas, digging into the experiments, sharing experiences in research. I would also like to thank Brais Martinez, who has helped me analyse experiment results and polished my work on knowledge distillation. My thanks also go to Jie Shen, who gave me important suggestions on the system design.

It goes without saying to present my appreciation to the Vice President's scholarship and support from the Computer Science Department. The Vice President's scholarship is very important to me during the pursuit of the PhD degree as it has directly solved my financial burden.

Last, but certainly not least, I want to dedicate this thesis to my parents for their unconditional support in my life. I would like to thank them for their love and guidance that are with me forever. I am also grateful to my other family members and friends who have supported me along the way. I also would like to thank friends during my PhD life: Jiankang Deng, Yujiang Wang, Pingchuan Ma and Jiuxi Meng for their encouragement, support and company to get through the stressful lockdown time.

Abstract

Face recognition has attracted particular interest in biometric recognition with wide applications in security, entertainment, health, marketing.

Recent years have witnessed rapid development of face recognition technique in both academic and industrial fields with the advent of (a) large amounts of available annotated training datasets, (b) Convolutional Neural Network (CNN) based deep structures, (c) affordable, powerful computation resources and (d) advanced loss functions. Despite the significant improvement and success, there are still challenges remaining to be tackled.

This thesis contributes towards in the wild face recognition from three perspectives including network design, model compression, and model explanation. Firstly, although the facial landmarks capture pose, expression and shape information, they are only used as the pre-processing step in the current face recognition pipeline without considering their potential in improving model's representation. Thus, we propose the "FAN-Face" framework which gradually integrates features from different layers of a facial landmark localization network into different layers of the recognition network. This operation has broken the align-cropped data pre-possessing routine but achieved simple orthogonal improvement to deep face recognition. We attribute this success to the coarse to fine shape-related information stored in the alignment network helping to establish correspondence for face matching.

Secondly, motivated by the success of knowledge distillation in model compression in the object classification task, we have examined current knowledge distillation methods on training lightweight face recognition models. By taking into account the classification problem at hand, we advocate a direct feature matching approach by letting the pre-trained classifier in teacher validate the feature representation from the student network. In addition, as the teacher network trained on labeled dataset alone is capable of capturing rich relational information among labels both in class space and feature space, we make first attempts to use unlabeled data to further enhance the model's performance under the knowledge distillation framework.

Finally, to increase the interpretability of the "black box" deep face recognition model, we have developed a new structure with dynamic convolution which is able to provide clustering of the faces in terms of facial attributes. In particular, we propose to cluster the routing weights of dynamic convolution experts to learn facial attributes in an unsupervised manner without forfeiting face recognition accuracy. Besides, we also introduce group convolution into dynamic convolution to increase the expert granularity. We further confirm that the routing vector benefits the feature-based face reconstruction via the deep inversion technique.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

With the rapid development of technology, biometric recognition software plays an increasingly significant role in modern security. Biometrics is the measurement and analysis of a human's distinctive physical or behavioural characteristics. Examples include but are not limited to fingerprint, retinal scanning, voice identification and facial recognition. Among them, face recognition has attracted particular interest because it's very easy to deploy. Face recognition aims at matching a given human face against a database of faces by measuring the distance of feature embeddings. Its applications have penetrated into various areas and here we categorize them into four broad domains: entertainment, health, marketing and security.

In **security**, recently released smartphones like Huawei, Xiaomi, Samsung, iPhone all launch a recognition-based authentication system to unlock phones. Compared with fingerprint authentication, the competitive advantage of face recognition is quick response and non-contact measurement. Such a system is also widely installed in various public places like railway station, airports, office buildings in China. It also assists officers in identifying and tracking criminals by comparing the suspect's face with the faces from surveillance camera systems.

In **entertainment**, face recognition is especially popular in social media. DeepFace from a research group in Facebook creates digital profiles from users and is used to identify human faces in digital images. Google photos employ the face recognition technique to sort pictures and automatically tag them based on the identification. FaceApp launched in IOS and Android systems provide the users to do some face editing works such as smile filter, hairstyle filter and age filter. These functions have added lots of fun to our daily life.

In **health**, face recognition has been used to diagnose diseases especially for those causing appearance changes. For instance, the National Human Genome Institute Research Institute employs face recognition to detect DiGeorge syndrome. Apple has launched two open-source frameworks ResearchKit and CareKit to assist clinical to monitor patients health remotely. Researchers at Duke University developed an Autism Beyond app that utilizes facial recognition-based algorithms to screen children for autism.

In **marketing**, face recognition can produce personalized recommendation after linking the browsing records and identity. This technique has been applied in shopping websites and video watching websites. For example, Alipay and WeChat in China also support face recognition for payment which brings much convenience in daily life as customers neither need physical bankcards nor the payment password.

Recent years have witnessed more and more mature face recognition techniques in both academic and industrial fields with a large amount of available annotated training datasets [46, 13, 195], Convolutional Neural Network (CNN) based structures [144, 51] and advanced loss functions [146, 34, 162]. Despite the significant success, there were still challenges remaining to be tackled (this is detailed in Section 1.2). It is worth mentioning and emphasizing here that in this thesis we seek to address and solve the following problems/cases:

1. A novel "FAN-Face" framework to improve the unaligned face recognition performance by gradually integrating features from a pre-trained facial landmark localization network into a recognition network to learn face embedding;

2. Advocate for a new knowledge distillation method via softmax regression representation learning by optimizing the output feature of the penultimate layer of the student network;

3. Interpret inner face recognition model behaviours by clustering and analysing visual attributes from a dynamically-routed CNN framework.

1.2 Problem Definition

The main focus of this thesis is to study the deep face recognition problem in the wild from three perspectives: performance, size, interpretability.

Performance: Through the investigation of the literature in Section 2.1, we notice that most algorithms share a conventional data pre-processing step. Align-cropping the face image based on predicted facial landmarks from the face alignment model before feeding them into the feature extraction model. One might be curious whether we can train a model on images without the align-cropped step and make it work well or better than the aligned counterparts. We believe this is a problem worth exploring as align-cropped step inevitably lose face areas, especially for profile faces. The lost information will not be compensated by adjusting the network structure, training strategies, and loss functions and definitely have a negative impact on recognition. Therefore, the first step of this thesis is to answer this question. The advantage of normalized face input has already been studied in the literature. Researchers have reached a consensus that aligning faces to a unified distribution can remove the appearance variations from rigid transformation and thus force the model to focus on learning identity-specific representation. The widely applied approach is applying a 2D affine transformation to calibrate facial landmarks to the predefined templates [34, 162, 94, 160]. So a key challenge in unaligned face recognition is the appearance variations caused by irregular landmark locations in the face region. Face

alignment aims at locating facial landmarks in face images and memorizes rich location related knowledge throughout the network. Therefore, a solution is to compensate for the missed location prior by borrowing knowledge from a pre-trained face alignment network. However, direct feature combination by addition or concatenation may hinder the optimization as the different features are captured by alignment and recognition networks. For instance, feature from alignment task is more shape focused while the feature from recognition task is designed to be robust to intra-class appearance variations.

Size: Three key factors (network design, training dataset, loss) contribute to superior performance on the face recognition task. Normally, a high capacity heavy model is trained on a large annotated dataset like MS1M [46] with an advanced classification loss like CosFace [162], ArcFace [34] and CurricularFace [61]. The obtained model is able to achieve good performance on the public benchmarks such as LFW [59], CPF-FP [136], AgeDB [104], IJB-B [171], IJB-C [100], MegaFace [71]. The whole procedure is time-consuming and computation resource hungry due to the large dataset size and parameters to be optimized. However, the face recognition system is designed to be deployed on devices with limited resources such as mobile phones. It is a challenge to deploy these cumbersome deep models due to their high computational complexity and large storage requirements. Given these real-world resource constraints, model efficiency has become increasingly important in face recognition. Also, it is hard to curve away from the prerequisites like large datasets and advanced loss. To tackle this problem in general object classification, a variety of model compression and acceleration methods have been studied including network pruning [48, 80], network quantization [125, 174], knowledge transfer/distillation [55, 204], and neural architecture search [229, 90]. The second step of this thesis is to apply the current state-of-the-art knowledge distillation approaches to face recognition task, systematically examine various choices, and then propose a novel approach to improve current performance. The challenge of this problem is how to optimize the face recognition accuracy to be optimal on the target task.

Interpretability: The last step is to answer "what is learned in the face recognition model". This step is to interpret networks' inner behaviours. Despite the significant advances in the performance of face recognition by designing network structure or advanced losses, there is a strong demand to interpret its behaviours, in order to understand, improve, and trust its decisions. However, so far there are few attempts to understand what is learned inside the network parameters and most are on the general classification task. Besides, recent interpretable CNNs observe degraded recognition accuracy after introducing interpretability. This drawback largely limits its practical applicability. The challenge of this problem is to define and analyse the interpretability of face recognition without deteriorating the model performance.

1.3 Contributions

In this section, we list three contributions made in this thesis.

- **"Fan-Face Network":** The first contribution is to improve unaligned deep face recognition with shape prior information provided by a pre-trained facial landmark localization (FAN) network. Both landmark heatmaps and features from the FAN are integrated into the face feature extraction process to (a) provide facial pose-, expression-, and shape-related information, and (b) help establish correspondence for improving face matching. Moreover, we have explored various architectural design choices at a network level to identify the best strategy for integration and, proposed a novel feature integration layer which is able to effectively integrate the features from the two networks. We show that such a simple approach systematically improves recognition on the public face recognition datasets, setting a new state-of-the-art on LFW [59], YTF [173], IJB-B [171], IJB-C [100] and MegaFace [71] datasets.

- **"Softmax Regression Representation Learning":** The second contribution is to advocate for a knowledge distillation method that optimizes the output feature of the penultimate layer of the student network and hence is directly related to representation learning. The final losses are composed of two parts: (a) L_2 loss to do direct feature matching between teacher and student's outputs; (b) softmax regression loss to let pre-trained teacher's classifier evaluate students' representation. The key novelty is from the second loss which considers the inter-channel dependencies and links feature representation with target accuracy. Moreover, we also extend our method to learning from unlabeled data with pretrained teacher trained on labeled datasets only to boost the student performance and further validate its superiority and generality in the classification task. Our method is extremely simple to implement and straightforward to train and is shown to consistently outperform previous state-of-the-art methods over a large set of experimental settings including different (a) network architectures, (b) teacher-student capacities, (c) datasets, and (d) tasks, (e) domains.

- **"Unsupervised Mixtures of Experts":** The third contribution is to interpret face recognition from learned visual attributes. The whole structure is a dynamically-routed CNN trained with a supervised face recognition loss and an unsupervised clustering loss. We observe that routing vector in dynamic convolution is capable of partitioning the data based on representational attributes but results in a noisy and scattered distribution. Based on these observations, we instantiate a group dynamic convolution to lengthen routing vector's dimension and in turn enhance its discrimination capacity. In addition, we propose to leverage an unsupervised clustering loss to push the routing vectors be distributed over a fixed number of clusters in the feature space without damaging the recognition performance. Due to the lack of annotated data with desired facial properties, we annotate three datasets with 5 facial attributes (gender, age, race, pitch and yaw) and then explore routing vectors' behaviour on them with clustering analysis. We demonstrate that learned clusters are able to encode rich facial properties without requiring hand-designed attributes

for supervision. Moreover, we present convincing results on the face inversion task that the routing vector provides auxiliary prior beneficial to producing images faithful to input.

1.4 Outline

We organize the rest of this thesis as follows:

- **Chapter2** reviews existing methods that are related to our thesis, including face detection, face alignment, face recognition, knowledge distillation methods and explainable artificial intelligence (AI). We introduce the representatives of each category to outline the main trend.

- **Chapter3** presents a novel architecture"FAN-Face" for unaligned face recognition via establishing shape-semantic correspondence among faces. We have detailed the various architectural design choices to identify the best strategy for inner feature combination and heatmap integration. We also examine SOTA alignment networks like ResNet, HRNet to validate the generalization ability of the proposed framework. Final experiments are done on various benchmarks with both verification and identification measurements to prove the effectiveness of the proposed method.

- **Chapter4** introduces a novel knowledge distillation method for model compression. We describe the two components: L2 feature matching and softmax regression representation learning. We have explored variants of the proposed method and analysed the effectiveness of the proposed method from different aspects such as transferability, clustering quality. We inspect how to improve students' performance from extra unlabeled data with the prior provided by teacher network trained on labeled data. To this end, we have explored and adjusted some popular tricks from semi-supervised learning into the usage of unlabeled data. The final comparisons are conducted across different tasks after training on labeled data alone and both labeled and unlabeled data .

- **Chapter5** proposes to interpret the inner behavior of face recognition model. We describe how to design an explainable structure with dynamic convolution and how to increase its expert granularity. Then, we examine two popular techniques in unsupervised clustering learning: DeepCluster [15] and Sinkhorn-Knopp [28]. Next, we investigate the training strategy to improve the interpret-ability without deteriorating recognition performance. Besides, we also describe how to obtain datasets with facial attributes. Finally, we quantitatively and qualitatively compare various variants in clustering analysis and feature-based face reconstruction.

- **Chapter6** concludes this thesis and discusses the future research directions of deep face recognition in the wild.

1.5 List of Publications

Below is a list of publications (and the corresponding chapters):

- Adrian Bulat*, **Jing Yang***, and Georgios Tzimiropoulos. “To learn image super resolution, use a gan to learn how to do image degradation first". In *Proceedings of the European conference on computer vision (ECCV)*, 2018. (* denotes equal contribution).
- **Jing Yang**, Adrian Bulat, and Georgios Tzimiropoulos. “Fan-face: a simple orthogonal improvement to deep face recognition”. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. **Chapter3**
- Brais Marinez, **Jing Yang***, Adrian Bulat*, and Georgios Tzimiropoulos. “Training binary neural networks with real-to-binary convolutions”. In *International Conference on Learning Representations (ICLR)*, 2020. (* denotes equal contribution).
- **Jing Yang**, Brais Marinez, Adrian Bulat, and Georgios Tzimiropoulos. “Knowledge distillation via softmax regression representation learning”. In *International Conference on Learning Representations (ICLR)*, 2021. **Chapter4**
- **Jing Yang**, Adrian Bulat, Keerthy Kusumam, and Georgios Tzimiropoulos. “Interpretable Face Recognition via Unsupervised Mixtures of Experts”. In submission. **Chapter5**
- Bulat, Adrian, Shiyang Cheng, **Jing Yang**, Andrew Garbett, Enrique Sanchez, and Georgios Tzimiropoulos. “Pre-training strategies and datasets for facial representation learning”. *arXiv preprint arXiv:2103.16554* (2021).

Chapter 2

Literature review

2.1 Components of Face Recognition

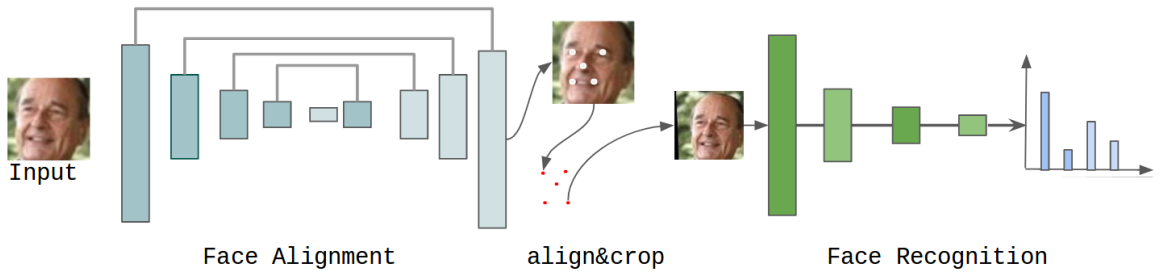


Fig. 2.1 Overview of face recognition. It consists of face localization and face feature embedding. The white dots are detected 5 landmarks. The red dots forms the mean face which is composed of left eye center, right eye center, nose tip, left corner of the mouth, and right corner of the mouth. The white landmarks are aligned to red landmarks to crop the normalized face area.

Figure 2.1 illustrates the pipeline of face recognition, mainly consisting of two components: face localization and face feature embedding. Face localization contains face detection and facial landmark localization. Firstly, a face detector is used to locate the faces in a given image. Then, a facial landmark detector is to predict the semantic landmarks of the detected face. Usually, five landmarks containing left eye center, right eye center, nose tip, left corner of the mouth, and right corner of the mouth are collected. With the estimated landmarks, the face can be aligned and cropped based on the normalized canonical coordinates from a mean face shape. In the literature, the advantage of the aligned crop operation is attributed to reducing appearance variances and making the subsequent feature embedding part more identity-focused. Finally, the align-cropped face regions are fed into the embedding model with the aim of encouraging features from the same identity to be close while the different ones to be far apart. In the following sections, we will describe each component in details as below.

2.1.1 Face Detection

Face detection, aiming at locating face regions in an image, is a fundamental stage for face-related tasks like face alignment, face recognition, facial action unit recognition *et al.* , and its

performance significantly determines the efficiency of the subsequent tasks. Recently, motivated by the rise of CNNs, the performance of face detection achieves considerable improvements and many milestone works have been proposed [84, 210, 20, 219, 106, 35]. Based on the core technical contributions, current face detection methods can be divided into three categories: cascade-CNN based models, region-based CNN (R-CNN) based models and single shot detector models. In the following part, we will introduce each category by describing its representatives.

Cascade-CNN Models

The core of cascade models is to train a sequence of small models to classify face and non-face regions in different stages to reject non-face areas from coarse to fine.

Li *et al.* [84] is one of the early CNN-based face detection models under cascade framework. In this work, the whole structure is formed by six CNNs with half of them doing face vs. non-face regions classification and the rest for bounding box calibration. To reject non-face regions effectively, those operations are conducted at multiple resolutions . To be specific, the detectors quickly reject the background regions at the former fast low-resolution stages. Then, the calibration stage improves the localization quality by reducing the number of candidate regions to subsequent classifier. Therefore, at the later high-resolution stages, the detector can carefully evaluate the remaining candidates. Compared with state-of-the-art methods at that time, the final obtained face detector is efficient and well-performed with running speed 14 FPS on CPU for typical VGA image as input and 100 FPS with GPU acceleration.

As an improved version of [84], Zhang *et al.* [210] have observed two drawbacks in [84]: the extra computational expense in the calibration stage and ignored correlation between face detection and alignment. To solve both issues, Zhang *et al.* have adopted and modified the cascade framework to jointly train the face detection and alignment in a multi-task manner. Additionally, they have introduced a new online hard sample mining strategy to automatically improve the performance without manual sample selection.

It has been shown that deeper layers are good at discriminating difficult face vs. non-face than shallow layers. Thus, Zhang *et al.* [211] have proposed “Inside Cascaded Structure” to do the binary classification task at different layers within the same CNN. The main improvement comes from two designs. On the one hand, a two-stream contextual CNN architecture is proposed to explore the body part information. On the other hand, a data routing mechanism following the “divide and conquer" idea is introduced to assign samples to various layers and thus let the deeper layer focus on harder samples.

In conclusion, the advantage of cascade CNN models is that they have a good trade-off between speed and accuracy as the former classifiers at earlier stages can shrink the background while keeping the faces with low computation cost. However, the computation complexity in the inference stage will be increased significantly when dealing with an image with lots of faces.

R-CNN based Models

The R-CNN based detectors mainly refer to the face detection models built on the generic object detection algorithms like Fast R-CNN [42], Faster R-CNN [127], R-FCN [29], and Mask R-CNN [50].

Chen *et al.* [20] have proposed a supervised transformer network for face detection. The whole framework is made up of two stages: multi-task region proposal network (RPN) and R-CNN. Motivated by the effectiveness of joint optimization, the RPN part is to jointly predict both candidate face regions and facial landmarks. The R-CNN in a later stage is to do further verification with the warped face candidate region as input. The whole framework is trained end-to-end to capture coarse-to-fine features. Finally, this model achieves 30 FPS running speed on a single CPU core when fed with a VGA-resolution image.

Jiang *et al.* have [67] directly applied the Faster R-CNN to the face detection task. Without considering the differences between human faces and generic object, pure Faster R-CNN is still able to achieve impressive face detection performance after retraining the parameters on the face detection datasets. This success is attributed to the effectiveness of the RPN module.

Another similar extension can be found in Face R-CNN [161]. Wang *et al.* have borrowed Faster R-CNN into face detection task and further boosted its performance by taking the characteristics that resided in the face detection task into considerations. In particular, they have introduced the center loss to enhance the intra-class compactness in the face vs. non-face classification. In addition, they have explored online hard example mining to search the appropriate setting between positive and negative hard samples. They have also designed a multi-scale training strategy. As a subsequent work [143], more techniques including feature concatenation, hard negative mining, multi-scale training, model pre-training, and proper calibration of key parameters have been analysed and explored in face detection under the Faster R-CNN framework.

Motivated by the success of region-based CNNs in the object detection and face detection tasks, CMS-RCNN [227] has been designed to combine both the region proposal component and the region-of-interest detection component to enhance the model's performance. Except for the contribution from the network design, Zhu *et al.* [227] have further explored the multi-scale information to locate the tiny face and contextual reasoning to reduce the overall false positives.

Single-shot Detector Models

The basic frameworks for single shot detector include SSD [93] and RetinaNet [88]. Different from the two-stage detectors based on the R-CNN framework, single-shot detectors finish face location after RPN without the subsequent R-CNN. Another advantage of the single-shot framework is the consistent computational complexity in inference while the two-stage one is seriously affected by the face number.

S3FD [219], full name as single-shot scale-invariant face detector, is a popular method under the SSD framework and it works especially well on detecting small faces after introducing three

tricks. Firstly, a scale-equitable face detection framework is presented to handle the various scales of faces. Secondly, a scale compensation anchor matching strategy is explored to increase the recall rate of small faces. Thirdly, a max-out background label is introduced to reduce the false positive rate of the small faces. S3FD has effectively surpassed the anchor-based detectors on detecting small faces.

FaceBoxes [218] is a CPU real-time face detector under an SSD framework with a lightweight yet powerful network structure. It consists of the Rapidly Digested Convolutional Layers (RDCL) and the Multiple Scale Convolutional Layers (MSCL). Specifically, the RDCL is to accelerate the inference speed on the CPU device while the MSCL is to handle face scales by enriching receptive fields and distributing anchors over different layers. In addition, a new anchor identification strategy is designed to balance the intensity of anchors across different layers. The empirical results have validated the effectiveness of FaceBoxes in detecting small faces. Another lightweight face detector is proposed in EXTD [202]. Yoo *et al.* have presented to iteratively reuse a shared lightweight and shallow backbone network to reduce the model size while preserving the model's performance.

SSH [106], as a single-stage headless face detection, is fast in speed and lightweight in model size. On the one hand, it is headless by removing the full connected layer classification part. On the other hand, it is scale-invariant by detecting multi-scale faces from different layers in a single forward pass of the network. Besides, SSH has proposed to add filters on each prediction head to enlarge the receptive field and merge the context information.

More recently, RetinaFace [35] performs pixel-wise face localisation on various scales of faces by taking advantages of jointly training supervised face detection, extra-supervised facial landmark regression and self-supervised face regression in a multi-task manner. During the training period, the three tasks benefit from each other, leading to a dramatic accuracy increment for face detection. Also, the main structure is a lightweight backbone designed based on feature pyramids. Therefore, RetinaFace can run real-time on a single CPU core for a VGA-resolution image.

2.1.2 Face Alignment

Face alignment aims at locating semantic facial landmarks in a given face image. It has wide applications in computer vision tasks such as face recognition, face emotion recognition, facial action unit recognition *et al.* . As a pre-acquisition step for face analysis, its performance plays a fundamental role in the subsequent tasks. In the literature, the face alignment methods are categorised into coordinate regression models and heatmap regression models. Recent advances in face alignment are largely attributed to the rise of deep neural networks.

Coordinate Regression-based Models

In coordinate regression-based models, model directly predict the facial landmark coordinates (e.g. x_1, y_1). Roughly, we can divide related methods into hand-craft features based cascade

shape regression and deep feature representation based coordinate regression based on feature extractor.

The main idea of cascade shape regression is to learn a sequence of regressors in an additive manner to approximate the mapping function from the initial mean shape to the ground-truth shape. At each iteration, the handcraft features like HOG, SIFT are extracted around the current predicted landmark, and the linear mapping is learned to reduce the distance gap between the current shape position and the target shape position. In the literature, this mapping solution can be founded by the least square algorithm.

Based on the above description, it is easy to conclude that the shape-indexed features play a critical role in the model performance. Thus some works focus on informative feature learning. Xiong *et al.* [186] have extracted the SIFT features around each landmark to learn the regression matrix via linear regression. Cao *et al.* [14] have proposed an explicit shape regression method on feature correlation selection. Ren *et al.* [126] have explored a set of random forests to learn discriminative binary features at local landmark region and finally obtain a model running at 3000 FPS.

Another key point is about the initial shape. Cascade shape regression works in a data-driven manner. It first attempts to remember the fitting paths in the training datasets and then directly utilizes these mapping projections in the testing phase. Therefore, when the assigned initial shape is far away from the final prediction, it has difficulty doing correction based on the noisy shape-index features. Therefore, Deng *et al.* [36] have proposed multi-view cascade shape regression in different pose subsets. Such a "divide-and-conquer" trick has significantly improved the model's robustness on the profile faces. This idea is further extended in facial landmark tracking in [192].

Thanks to the development of neural network and its wide applications in the computer vision field, the coordinate regression based on deep features have been explored and gained significant performance improvement in face alignment benchmarks. These methods are further categorised into local-based and global-based models. Local-based methods can be seen as an extension of the shape-index feature on the deep feature. For instance, [155] has extracted features on the neighbourhoods of all landmarks and then feeds local features into regression CNN to do coordinates prediction. However, the local shape-index deep feature still meets the problem of being sensitive to initial estimates of facial landmarks or the assigned mean shape.

In contrast, for global-based methods, an image patch enclosing the whole face region is fed into the regression CNN to predict 2D coordinates of facial landmarks. The advantage of the global-based methods is that it does not require the initialisation of facial landmarks.

In [145], the whole framework is composed of three-level carefully designed networks. Motivated by the process that the entire image as input and key points as output, both the entire input's texture context information and implicit encoded geometric constraints are explored and utilized to locate each key point. Finally, the obtained model is more robust to factors like occlusions, large pose variations, and extreme lightings.

Different from designing different deep neural networks to improve the model's performance with fixed L2 loss, Feng *et al.* [41] have defined a loss function named Wing loss to increase the

contribution from samples of small and medium-size errors in network optimization. Extensive experiments have been conducted under several CNN architectures on popular benchmarks to verify the effectiveness of the proposed Wing loss function.

Heatmap Regression-based Models

In addition to the coordinate regression, recent heatmap regression based on advanced neural network designs have shown promising results. Different from coordinate regression, the network's prediction is an implicit confidence map of each landmark instead of the direct x,y coordinates on the input image. The ground-truth heatmaps are generated from extra operation in the data loader by setting a Gaussian circle around each landmark with the highest confidence at center. [193] is the pioneering work to bring hourglass network performed on human pose estimation to facial landmark prediction, and such direct application have achieved superior performance over other methods at that time. At a similar time, Bulat *et al.* [10] have replaced the basic block with a more advanced multi-scale block to capture more information within one block, and have gained further performance. It is worth mentioned here that this work has been further extended to the binary face alignment model with decent performance [8].

Tang *et al.* [150] have designed stacked U-Nets to reuse features shared similar resolution across different layers. To further enhance the performance, an order-K coupling design together with iterative refinement and memory sharing mechanism are explored in training. Finally, the gained model achieves state-of-the-art performance with 70% fewer parameters. A similar idea can be found in [149].

So far, the above-introduced heatmap regression models are all driven by minimizing L2 distance loss between the prediction and ground truth. Motivated by the Wing loss's success on coordinate regression [41], [164] have proposed an adaptive wing loss on confidence maps which helps to adapt current shape flexibly according to ground-truth heatmap pixels. The experimental results have shown that the foreground pixels are penalized more when compared with the background ones. To address the imbalance distribution among different locations, Wang *et al.* have proposed a weighted loss map to guide the optimization emphasis on crucial landmarks on the foreground and difficult background pixels.

By employing extra information from the boundary, Wu *et al.* [176] have proposed to use boundary lines as the geometric structure of a human face to remove the ambiguities in the landmark definition.

More recently, [78] has started to analyse uncertainty and visibility of predicted landmarks and has presented a novel end-to-end trainable framework called LUVLi to simultaneously estimate the landmark position together with the uncertainty and visibility. Benefiting from the side information provided by the uncertainty and visibility, the final model has achieved superior performance on several datasets. The empirical results have proved that without any supervision, the model is still able to distinguish between clear and externally occluded landmarks. In addition, the model has also replaced the argmax with the spatial mean of the ReLU heatmap to obtain sub-pixel accuracy.

2.1.3 Face Recognition

Face recognition is a fundamental task in the field of pattern recognition and machine learning. The target of face recognition consists of face identification and face verification. Face identification is to assign a given face to a specific identity while face verification is to determine whether the two given faces are from same identity. Recent face recognition methods are built on deep discriminative features with advanced loss functions. These loss functions can be divided into pair-based loss and margin-based softmax loss.

Pair-based Loss

The contrastive loss [147, 146, 196] and the triplet loss [134, 116, 92] are the seminal examples of loss functions for pair-based loss [184, 169] in deep metric learning.

The visual representation of a gradient descent step for contrastive loss is in Figure 2.2. In practice, it takes the embedding vectors from a face image pair and aims at pulling them together in Euclidean distance if they are from the same identity (positive pair) and pushing them apart if they are from the different identities (negative pair). The contrastive loss is formulated as:

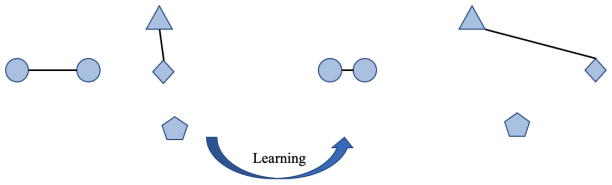


Fig. 2.2 Contrastive loss. The goal is to pull features from same identity together and push features from different identities apart. Different shapes denote different identities.

$$\begin{aligned} L =& I_{ij} max\left(0, \left\|x_i - x_j\right\|_2 - \varepsilon^+\right) \\ &+ (1 - I_{ij}) max\left(0, \varepsilon^- - \left\|x_i - x_j\right\|_2\right), \end{aligned} \tag{2.1}$$

where $I_{ij} = 1$ indicates x_i and x_j are positive pair and $I_{ij} = 0$ indicates negative pair. x is the feature embedding. ε^+ and ε^- represent the the margin for positive pairs and negative pairs respectively. DeepID2 [146] has combined softmax loss and contrastive loss together to obtain discriminative face embedding for face recognition. This work is further extended in DeepID2+ [147] and DeepID3 [144] by boosting performance with advanced network designs.

The visual representation of a gradient descent step for triplet loss is in Figure 2.3. In practice, it learns a discriminative feature embedding by minimizing the distance between an anchor and a positive sample and maximizing the distance between the anchor and a negative sample in a triplet. The triplet loss is formulated as:

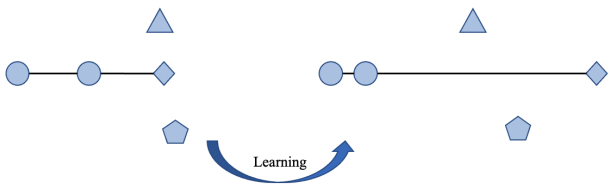


Fig. 2.3 Triplet loss. The goal is to minimize anchor's distance to its positive sample and maximize the distance to its negative sample. Different shapes denote different identities.

$$L = \left\|x_i^a - x_i^p\right\|_2^2 - \left\|x_i^a - x_i^n\right\|_2^2 + \varepsilon, \tag{2.2}$$

where x_i^a, x_i^p, x_i^n denote anchor, positive sample from same identity with anchor and negative sample from a different identity with anchor respectively. ε is a margin and x is the feature embedding. FaceNet [134] has used triplet loss to learn feature embedding in Euclidean space and further explored different triplet selections to boost model's performance. Similar idea can be found in [134, 116, 92]

Pair-based losses benefit from a large amount of sample-to-sample comparison via aggressive tuple sampling during training. However, that tuple number increasing polynomially with training data size causes the prohibitively high training complexity and instability. Besides, not all tuples contribute to training and ineffective tuples even incur model degradation. Therefore, most pair-based loss methods will explore the sampling techniques to enhance the useful pairs in training. However, this operation will introduce extra hyperparameters and need more efforts and experience.

Margin-based softmax Loss

Face recognition task in training step can be treated as a multi-class classification problem including two parts: feature extractor and classifier. The difference is only in testing. For the general classification task, a classifier is used to predict the input's label. However, in face recognition, only the former part is maintained to do feature extraction and comparison while the latter part is deleted. Thus softmax loss as the most classical loss in classification can also be explored in face recognition and it is formulated as follows:

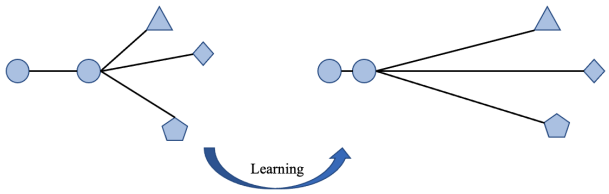


Fig. 2.4 Margin-based softmax Loss. It is a variant of softmax loss by adding the margin on the cosine distance between current sample and its prototype assigned by label. Different shapes denote different identities.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{k=1}^{K} e^{W_j^T x_i}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{k\neq y_i}^{K} e^{W_j^T x_i} + e^{W_{y_i}^T x_i}}, \tag{2.3}$$

Here, w denotes the weight of classifier, where $k \in 1,2,...,K$ denotes class number. x_i is the embedding feature of $i-th$ sample. N is the batch size. We omit the bias here for simplicity. In particular, in face recognition, we normalize both the weight vector in the classifier and feature embedding. Thus the calculation between weight vector and embedding feature evolves as measuring the cosine similarity and Eq.2.3 can be rephrased as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s*cos(W_{y_i}^T, x_i)}}{\sum_{k\neq y_i}^{K} e^{s*cos(W_j^T, x_i)} + s*e^{cos(W_{y_i}^T, x_i)}}, \tag{2.4}$$

Here, s is for keeping value magnitude [160]. From the similarity aspect to explain Eq.2.4, it aims to increase the similarity between x_i and positive prototype for class i and decrease the

similarity between x_i and negative prototypes. Contrary to the sample-to-sample comparison in pair-based loss, here comparison is done between sample and prototypes. The recent five years have witnessed the new trend of margin-based softmax loss functions to boosting the face recognition performance. The losses are concluded in a general formulation as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{s*f(W_{y_i}^T,x_i,m)}{\sum_{k\neq y_i}^{K}s*f(W_j^T,x_i)+s*f(W_{y_i}^T,x_i,m)}, \tag{2.5}$$

Here, f is a margin function. The pioneer work, SphereFace [94], has proposed a multiplicative angular margin penalty $f(W_{y_i}^T,x_i,m) = cos(m*arccos(W_{y_i}^T,x_i))$ to enforce intra-class compactness and inter-class discrepancy simultaneously, leading to a better discriminative power of the trained model. Even though Sphereface [94] has introduced the important idea of margin into the softmax loss, the integer-based multiplicative angular margin makes the target logit curve very precipitous and thus hinders convergence. To this end, a hybrid loss function is further explored to stabilise training. Inspired by SphereFace [94], CosFace [162] has introduced an additive cosine margin $f(W_{y_i}^T,x_i,m) = cos(W_{y_i}^T,x_i) - m$. ArcFace [34] has introduced an additive angular margin $f(W_{y_i}^T,x_i,m) = cos(arccos(W_{y_i}^T,x_i) + m)$ with a well-formed geometrical interpretation. m in SphereFace is set as 4, in CosFace is set as 0.35, while in ArcFace it varies on datasets. More recently, FairLoss [89] and AdaptiveFace [91] have proposed adaptive margin based on classes to address long-tail distribution in training dataset.

The visual representation of a gradient descent step for margin-based softmax Loss is in Figure 2.4. Compared with paired-based loss, margin-based softmax loss is easy to train with better performance. However, as is mentioned in [159], it is vulnerable to label noise in the dataset. To increase the robustness, Sub-center ArcFace has [33] relaxed the intra-class constraint of ArcFace by designing K sub-centers for each class instead of one positive center.

2.2 Knowledge Distillation

Knowledge distillation has been widely adopted to compress the model with decent performance. In particular, it improves the performance of low-capacity, low-accuracy student network with the guidance from high-capacity, high-accuracy teacher network. Recently, it is an active topic especially in image classification task. The core of knowledge distillation is the definition of knowledge. Based on this, we can broadly categorise current methods into logits-based knowledge distillation [55, 26, 21], feature-based knowledge distillation [129, 204, 53] and relationship-based knowledge distillation [197, 82, 115]. In the following section, we will briefly introduce each category with individual representatives and its applications.

2.2.1 Logits-based Knowledge Distillation

In KD [55], knowledge is defined as the teacher's logits after the final softmax layer. The main idea is to train a student model to mimic the final prediction of a pre-trained and fixed teacher

model. The rationale behind KD is that the teacher's output carries richer information than a one-hot label because it encodes extra supervision signals about the inter-class similarities. For example, in the Mnist classification task, the teacher's prediction can encode the similarity between 7 and 9 while the one-hot label treats 9 the same as the other classes like 2,3 without considering the correlation.

A typical logits-based knowledge distillation model is illustrated in Figure 2.5. Suppose teacher's prediction as p and student's prediction as q, loss in [55] is defined as KL divergence loss between p and q:

$$L_{KD} = KL(q, p). \tag{2.6}$$

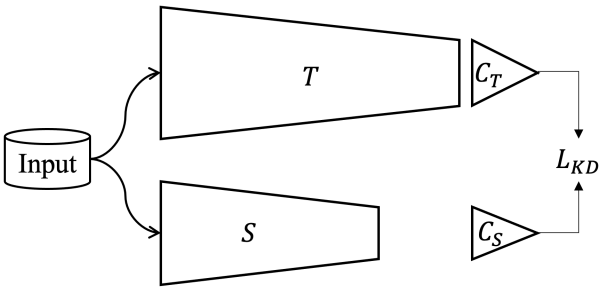


Fig. 2.5 The generic logits-based knowledge distillation. T, S, C denote teacher, student and classifier respectively. L_{KD} loss is added at the end of classifier's output.

There is a temperature parameter in the softmax function applied to p and q to keep the gradient magnitude and usually set as 4.

As a subsequent work of KD, [26] has shown experimentally that very accurate networks are "too good" to be good teachers and therefore has proposed to tackle this issue with early stopping technique in network training. This technique has especially improved KD's performance on the pair ResNet34 as a teacher and ResNet18 as a student in the ImageNet-1K datasets.

So far, the vanilla KD has a wide range of applications in the computer vision area. For example, in [21], Chen *et al.* have applied logits-based knowledge distillation to object detection under the Faster R-CNN framework. As is already described in Section 2.1, Faster R-CNN consists of region proposal network and region classification network. The classifiers from different stages are to distinguish the positive and negative anchors and specific classes respectively. It is a natural operation to apply the KD on the above two tasks with some task-specific modifications. Experiments have proven the effectiveness of KD when it is applied with various teacher-student pairs in the object detection task.

More recently, Zhang *et al.* [208] have applied KD to human pose estimation under the hourglass framework. Intuitively, the performance of the hourglass framework depends on the channel number (width of network), and the number of stacked hourglasses (depth of network). Therefore, a heavy network consisting of 8 hourglasses with 256 channel number is trained as a teacher to distill the knowledge to a lightweight student network composed of 4 hourglasses with 128 channels. Extensive evaluations on popular benchmarks have demonstrated the advantages brought by KD as the teacher's prediction provides additional more accurate supervision than annotation and helps to calibrate the mistake caused by the missing labels.

2.2.2 Feature-based Knowledge Distillation

Instead of defining the knowledge as the final output of the network, intermediate representations such as raw feature tensors [129] or processed attention maps [204] are also the origin of knowledge. This knowledge has been analysed and investigated to define loss functions to facilitate the optimization of the student by mimicking the teacher's intermediate representations. A typical feature-based knowledge distillation framework is illustrated in Figure 2.6.

In [129], knowledge is defined as the outputs of a teacher's different hidden layers. Romero *et al.* have discussed the two factors that affect the student's performance. One is the position to put the distillation. Intuitively, a pre-trained network captures the coarse to fine information of input with edge contour information in the earlier part of the network and abstract class-related information in the latter. Therefore, incorrect distillation position will incur over-regularization in training student. To tackle this issue, Romero *et al.* have chosen to put the distillation position in the middle of the network to avoid constraining the flexibility from the guided layers. The other is to deal with incomparable tensor size as there is no guarantee that features from the student network and teacher network share the same shape. Therefore, Romero *et al.* have proposed to use a regressor network to adjust the student feature size. This regressor network will not add extra complexity in inference as it will be discarded in the testing period. Finally, the MSE loss is introduced to optimize student learning. The hint-based training has suggested that new research direction is to explore advanced training strategies to leverage the power of deep networks.

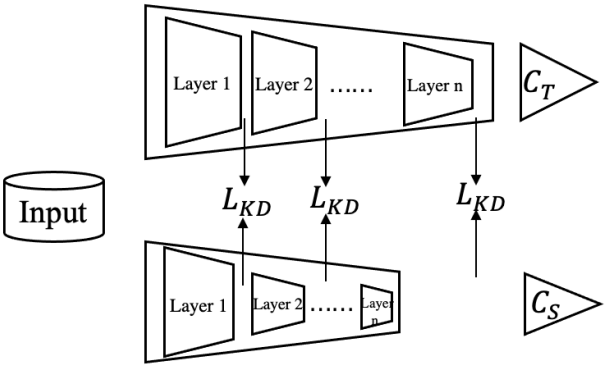


Fig. 2.6 The generic feature-based knowledge distillation. T, S, C denote teacher, student and classifier respectively. L_{KD} loss is added at different locations inside the backbone.

As is analysed in FitNets [129], matching the whole feature tensor is hard and in certain circumstances will adversely affect the performance and convergence of the student. To relax the assumption of FitNets, Attention Transfer (AT) has been proposed in [204] where knowledge takes the form of attention maps which are summaries of the energies of the feature tensors over the channel dimension. Specifically, knowledge in AT is defined as a spatial attention map in a teacher's hidden layer. The authors have observed that the attention map contains information about where to focus in classifying an object. The authors have defined three types of attention across channels: (a) sum of absolute values; (b) sum of absolute values raised to the power; (c) max of absolute values raised to the power. Finally, the loss is calculated as the MSE distance between the teacher's attention map and the student's one. Also, to solve the spatial size incompatibility issue, AT has proposed to use an adaptive pooling layer to adjust the size before calculating the attention map. AT has claimed that it is the first work that works on ImageNet-1K

[77] in pair of ResNet34 as teacher and ResNet18 as a student. An extension of [204] is to use Maximum Mean Discrepancy of the network activation and has been presented in [62].

Recently, [53], as a subsequent work of [54], has proposed a new feature distillation loss after a comprehensive investigation and measurements on fours factors: teacher transform, student transform, distillation position, and distance function. Heo *et al.* have suggested putting the position of the distillation between the first ReLU and the end of layer block and have proposed a specifically designed distance function called partial L_2 to transfer only the useful (positive) information from the teacher to the student after margin ReLU transform.

More recently, motivated by the success of Neural Architecture Search (NAS), Li *et al.* [83] have proposed to supervise the block-wise architecture search by the architecture knowledge distilled from a teacher model. Li *et al.* have observed that the knowledge not only lies in the network parameter but also lies in the network architecture. Another NAS based method has been proposed in [45], in which a student-to-teacher loss is utilized to find the aggregation weights that match the learning ability of the student under a two-stage framework via differentiable aggregation search.

Passalis *et al.* [118] have claimed that traditional knowledge distillation methods ignore information plasticity during the training process, which causes performance degradation in training significant light-weight student. The authors have proposed to model the information flow through the various layers of the teacher by introducing an auxiliary model based on a critical learning scheme. The experiment has shown that [118] works especially well on teacher-student pair with different architectures which is a barrier in other frameworks.

2.2.3 Relation-based Knowledge Distillation

So far, both the logits-based knowledge and feature-based knowledge is a 1:1 corresponding relationship. That is to say, for one input, a student is to mimic the teacher's behaviour on its input. Relation-based knowledge distillation observes that such mimic behaviour fails to consider the relationship among inputs and have proposed to focus on exploring transferring the relationship between features rather than the actual features themselves. A typical relation-based Knowledge distillation framework is illustrated in Figure 2.7.

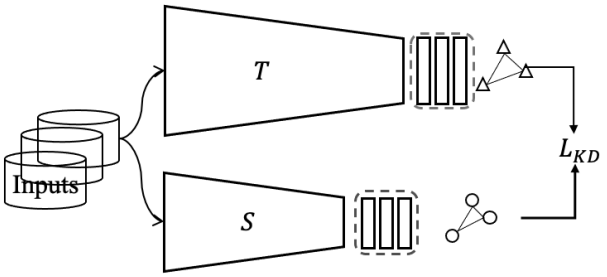


Fig. 2.7 The generic relation-based knowledge distillation. T, S, C denote teacher, student and classifier respectively. L_{KD} loss is added on the feature embedding extracted after backbone.

In [197], knowledge is defined as feature correlations across different layers. After computing the Gram matrix of features across layers for both teacher and student, an L_2 loss is applied on pairs of teacher-student Gram matrices. By doing so, flow between different layers among teacher can be learned by the student. Experiments have been done on the transferring task to validate the effectiveness of the proposed method. The limitation of this work is the high

computational cost which has been further addressed in [82] by compressing the feature maps via singular value decomposition.

In [115], knowledge is defined as the mutual relationship within a batch. Park *et al.* have proposed a relational knowledge distillation (RKD) method. For concrete realizations of RKD, they first computed distance-wise and angle-wise relations between each embedded feature vectors and then enforced structural similarity for student and teacher. Similar idea is further explored in [120] and [95]. In [120], Taylor series expansion has been used to better capture the correlations among multiple instances. Several kernel tricks have been explored to capture knowledge in high-dimension feature space. Besides, sampling strategies are further investigated to better consider the inter-class and intra-class correlation. The experiments are done on object classification, person re-identification and face recognition to validate the superior performance. In [95], the instance feature and its relationships to others have been considered as vertexes and edges respectively in a graph and the instance relationship graph has been optimized to model the feature space transformation across layers.

Inspired by the observation that semantically similar inputs should have similar activation patterns, [156] has proposed a similarity-preserving knowledge distillation method by guiding the student to mimic the teacher' activations. Specifically, Tung *et al.* have first extracted feature after different layers for the teacher and student network and then calculated the similarity matrix on a normalized feature map across spatial position to encode the correlation among different positions. Finally, MSE loss is applied to enforce the similarity between teacher and student's feature maps to drive the optimization of the student model.

More recently, [65] has proposed to match the student's output with the teacher's by distilling the knowledge through a quantized visual words space to concentrate more on important semantic concepts and spatial correlations. To this end, Jain *et al.* first used k-means to obtain a dictionary of feature representations in the training dataset and then trained the student to predict similar code as a teacher. The rationale behind this idea is that the dictionary can encode what is stored in feature space in the training dataset. For instance, it may capture some contour structures of a class.

Li *et al.* [86] have proposed the local correlation exploration framework to represent the relationships of local regions in the feature space which contains more details and discriminative patterns. This work argues that current knowledge distillation methods mainly pay attention to the sample-level feature and relationship consistency in global features while ignoring the details and discriminative patterns that exist in local features. Therefore, they have proposed to model the inter-class and intra-class sample correspondence in the same local position together with inter- correspondence across different local positions. By doing so, the student can produce more discriminative local features. Finally, a similar connection between distillation and representation learning has been very recently made in [153] which uses contrastive learning for knowledge distillation.

2.2.4 Self-distillation

There is a predefined teacher network trained on training datasets to guide the student in training in the above-mentioned methods. This kind of knowledge distillation is categorised as off-line knowledge distillation as the teacher network is pre-trained and fixed. Different from this procedure, self-distillation is to train a student network without an explicit teacher network, in which the deeper representation is to guide the shallow network in training. The generic overview is in Figure 2.8.

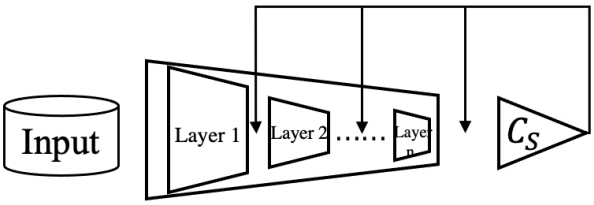


Fig. 2.8 The generic self-knowledge distillation. S denotes student network. L_{KD} loss is added between intermediate network output and final classifier's output.

As the pioneering work in self distillation, Zhang *et al.* [212] has defined the self-distillation concept. To be specific, Zhang *et al.* have proposed to improve the representation ability of feature maps extracted at different positions of the network. This idea is motivated by [58], in which the authors have found that directly adding classification loss after each layer in ResNet structure will deteriorate the whole network performance while this phenomenon is relieved in DenseNet. They have attributed it to the over-regularization in coarse layers. Therefore, [212] has focused on improving the ResNet performance by introducing auxiliary operations in different positions before adding classification loss. Finally, the final optimization has three sources: (a) the classification loss across different layers guided by cross-entropy loss with ground-truth labels, (b)hints loss across different layers guided by network output feature, (c) KL divergence loss across different layers guided by the final prediction. These losses are optimized together to improve the network's performance from coarse to fine layers. This work is further extended in their NeurIPS paper named SCAN [213] by improving auxiliary network design. In [213], the authors have observed that the third layer performance is worse than training the network alone. After digging into this problem, they have proposed an attention module as the auxiliary network and experimentally this design has enhanced the performance of shallow classifiers.

A similar idea can also be found in [122]. The multi-exit architecture proposed in this work has also shared the self-distillation framework. The knowledge in the final exit is employed to guide the training of early exits. Also, both cross-entropy loss on the ground-truth label and KL divergence on final output loss have been combined to optimize the network. Except that, they have proposed the temperature annealing in KL divergence loss to keep the teacher's output constant and the final experiments are done on unlabeled data to show the advantage of the proposed idea.

As an application of self-distillation and attention transfer [204], [56] has used the combination of two works to improve performance in the lane detection task. The observation is that attention maps in the pre-trained network have encoded rich contextual information.

Motivated by this phenomena, [56] has treated the attention map at the last layer as a form of "free" supervision to guide layer-wise attention distillation within the network.

As a special variant of self-distillation, Yang *et al.* [190] have treated the networks trained at earlier epochs as a teacher and designed a cyclic learning rate policy to transfer knowledge in earlier epochs to later epochs in the same generation within the same network. They have discussed three principles to achieve success including (a) the quality of teacher; (b) teacher-student difference; (c) secondary information.

Recently, [102] has first analysed theoretical support behind self-distillation Hilbert space. Mobahi *et al.* have shown that self-distillation works because it modifies regularization by progressively limiting the number of basis functions so that it can be used to represent the solution. They also have verified experimentally that the over-fitting issue can be reduced in a few rounds of self-distillation but the under-fitting issue alongside deteriorating performance will be caused after further rounds.

More empirical experiments can be found in [220]. The empirical evidence has shown that in the self-distillation framework, diversity of teacher predictions is closely related to student networks' performance. Inspired by both the observation and theoretical analysis, they have interpreted the knowledge distillation with amortized MAP and further proposed more efficient ways to obtain priors for instance-specific regularization in self-distillation.

2.3 Explainable AI

2.3.1 Network Visualization

Visualization of features from CNN is the most straightforward solution towards understanding neural networks and it also serves as a technical tool for diagnosing CNN representations in the literature with representative works. We can roughly category them into activation maximization based methods [40, 206, 103, 110] and class activation mapping based methods [135, 18, 139, 206].

Activation maximization [40] (a detailed review in [109]), as a representative visualization technique, is to find a solution that highly activates a given neuron by operating gradient descent optimization on the input. The rationale behind this idea is that a unit's maximum response to a pattern resides in its first-order representation.

Zeiler *et al.* [206] have taken both functions of the internal feature extractor and the operation of the final classifier into consideration. In practice, Zeiler *et al.* have exploited to locate which input pattern will cause a given activation in the feature map by mapping the stimuli back to the input pixel space. Another contribution of this work is that it has proven that the model's performance is closely related to the depth of the network rather than individual sections, which gives insights to effective network design.

Mahendran *et al.* [97] have explored the activation maximization idea by answering the question: is it possible to reconstruct the input image when given a feature representation

extracted from the network? Mahendran *et al.* have answered this question by inverting input on a given feature. The final solution can be found by optimizing the input with the gradient provided by reconstruction loss and regularization. As there is no unique solution to the inversion problem, the regularization here is to introduce image prior such as V^{β} norm to make generated images satisfied with human visual perception. This solution can be easily applied to both handcraft feature like HOG and SIFT and deep feature from CNN models. Similar ideas can be found in [103, 110].

More recently, the image inversion idea has been applied to the dataset reconstruction task. For example, in [198], the student network has been trained only with guidance from the teacher network without original training datasets. In particular, Yin *et al.* have synthesized the training dataset by optimizing random noise input to maximize the activation of the classifier in the teacher. By doing so, images can be generated according to the assigned labels from the teacher network. Besides, to improve the image quality, the information stored in batch normalization in the teacher network has been utilized to constrain the statistics calculated on generated inputs. In addition, Jensen-Shannon divergence between the teacher and student networks' logits has been introduced to improve the diversity of synthesized images. Experiments on CIFAR10 and ImageNet-1K have verified the effectiveness of the deep inversion idea.

Another line of network visualization is called class activation mapping (CAM). The core idea is to generate a heatmap image whose size is the same as input while the value is determined by its contribution to the final prediction. Intuitively, the larger value represents a higher response from the class label. The essence of CAM is to weighted sum the channel-wise representation in the selected layer. From the perspective of the weight sources, current CAM variants can be split into gradient-free CAM [226, 163, 105, 123] and gradient-based CAM [135, 18, 111].

In gradient-free CAM, [226] is the pioneer of CAM. The later part in the network of [226] consists of (a) global average pooling on 4-D feature maps to get a feature vector and (b) a fully connected layer to assign labels. The procedure of calculating the class activation map can be divided into 3 steps:

1. Extract the class weights from fully connected layer based on the assigned label;
2. Expand the class weights to share the same size as 4-D feature map to operate weighted sum on the feature map with class weights to get a heatmap;
3. Operate the ReLU activation on the heatmap to get the final normalized heatmap.

The obtained heatmap known as class activation maps allows us to locate the discriminative object parts detected by the CNN after drawing the predicted score on the input image.

Score-CAM [163] is an improved version of Grad-CAM [135]. Score-CAM has abandoned using the gradient from target class to compute weights. The authors have observed nuisances caused by gradients and mistakes from high confident samples. Instead, [163] has defined "increase of confidence" as the final visualization map which is achieved by subtracting the baseline score map from the gained class score map.

Another solution to cope with noise from gradients is proposed in SS-CAM [105]. The smoothGrad technique has been utilized to smooth the output noise. In this work, the authors have proposed two variants: (a) adding noise on the feature map (b) adding noise on the input. There is no clear conclusion about which one is better as the performance varies on the datasets.

Ablation-CAM [123] has conducted a thorough analysis on the importance of units in the feature map. By sequentially erasing the channel in the feature map and then comparing it with the uncontaminated one, it allows us to find the essential component in the model's performance. Finally, the gap in class score has functioned as the combination weights on the feature map.

For gradient-based CAM, in Grad-CAM [135], Selvaraju *et al.* have proposed a class-discriminative localization technique to interpret CNN based on visual explanations. Specifically, for a given target concept, for example, the "dog" category in the classification network, the gradient of this target would flow into any layer in the network to produce a coarse localization map. Compared with the previous CAM applied to specific structure design, this method is more general and can be applied to any layer.

Grad-CAM++ [18] is an extension of Grad-CAM by considering the contribution of different elements in gradient map. The extra weight can help to produce more precise localization and make it appliable to scenes with multi-class objects. Another work named smooth Grad-CAM++ [111] has introduced smoothGrad in gradient to alleviate the noise on saliency maps by averaging the gradients from multiple noisy inputs.

In addition, [139] has proposed to visualize image classification models by generating an image to maximize a class score and computing a class salience map to a given class, based on the gradient of the class score. [206] has introduced a multi-layer deconvolutional network to reveal the input stimuli that excites individual feature maps at any layer of the model. [38] has learned an up-convolutional network to predict input image from its feature vector and the obtained model is applicable directly in testing.

2.3.2 Pattern Detector

Pattern detector-related works attempt to connect the annotated visual patterns with inner filters in deep network to increase its interpretability. Some works focus on analysing the pretrained networks [225, 226] while the others resort to changing the network structure [5, 138, 19] or adding auxiliary loss [43, 216] to make the network more explainable.

Zhou *et al.* [225] have observed that units in inner layers function as visual concept detectors in the scene classification task. More specifically, the early layers in the network are responsive to low-level concepts like colours or textures while later layers tend to capture more high-level semantics like objects or scenes. Therefore, they have demonstrated that scene recognition task and object localization task can be done in a single forward-pass without being explicitly taught the notion of objects. This work is further extended in [226]. After an in-depth analysis on global average pooling and classification layer, Zhou *et al.* have observed that scene classification task works as a combination of object-specific units which are indeed stored in global average pooling.

Bau *et al.* [5] have proposed a network dissection framework to quantify the interpretability of inner representations by evaluating the correspondence between individual hidden units and selected semantic concepts. The experiments have validated that feature extracted at different positions can encode different categories of meanings and also training techniques will affect the interpretability of the representation.

Part detector discovery [138] has been proposed to locate semantic part-related activation centers based on computing and analysing gradient maps. This idea is further explored in [137], Simon *et al.* have treated the channels of a CNN as a part detector and then they have learned a part model in a completely unsupervised manner by selecting part detectors that fire at similar relative locations.

A similar idea is also taken in [43, 216], Gonzalez *et al.* have gone a step further by assisting each filter with a bounding-box regression loss. The auxiliary loss will help to refine the stimulus detection for each part class. Therefore, the conclusion can be made based on the phenomenon that whether the network's convolutional filters fire on semantic parts. Similarly, Zhang *et al.* [216] have designed a new loss to push filters in high Conv-layers to represent single parts of an object without additional annotations for supervision.

In [215], Zhang *et al.* have built a semantic And-or Graph on the pre-trained convolution neural network by associating certain inner units with the semantic part. By mapping the high-dimensional activations to low-dimensional representation concerning parts/sub-parts, the proposed framework is more robust and efficient in multi-shot learning. This idea is further extended in [214] to disentangle part patterns captured in each filter after considering the mixture of object parts existing in the conv layer. Zhang *et al.* [217] have learned a decision tree to analyse the decision made at the semantic level. Specifically, they have proposed to first use a decision tree to break down high-level feature into elementary concepts about object parts and then to analyse how the object parts fire filter and contribute to the final prediction.

2.3.3 Learning Interpretable Representations

CNN-based models often suffer from a lack of interpretability due to their non-linear nature. Optimizing with supervision loss from annotated labels only makes the models function as a "black box" in which only input and output are considered. To address this issue, some works focus on learning interpretable representation by adjusting or proposing a new network structure [85, 19, 166, 131].

Li *et al.* [85] have proposed an explainable image classification architecture consisting of an auto-encoder and a prototype classifier. Specifically, The encoder part maps the high dimension input to low dimension latent space and the decoder part transforms back to the input space. The prototypes are learned on the encoded input with cross-entropy loss. Except that, another two auxiliary losses constraining the correspondence between encoded inputs and assigned prototypes are trained together with supervised classification loss. In addition, the decoder part is guided by reconstruction loss from the input. Finally, the trained model is more interpretable as

prototypes provide useful cues to understand the inner workings within the classification network and can be visualized in a decoder network.

Unlike [85] where the prototypes are learned from the entire image, [19] has introduced a Prototypical Part Network to focus on the patch-based local representations. The rationale behind this work is that the network first understands the input with cues from prototypical parts, and then the final decision is made based on combined evidence from all local parts. This is quite similar to the human thinking mode. In addition to the supervision loss from the classification task, an extra constraint is added to enforce each convolutional filter to be identical to some latent training patch to make the filters interpretable in visualization. Both [85] and [19] are trained from scratch and thus faithful to input images.

Wang *et al.* [166] have developed a distillation guided routing method to effectively discover the critical routing paths for the input sample. Intuitively, the model's performance is determined by important channels' activation. If the important channels were suppressed in the routing paths, its performance would deteriorate severely. To this end, Wang *et al.* have introduced the knowledge distillation idea to off-line dissect the critical channels without damaging its performance. They have observed that the discriminative ability of routing nodes increases as the network goes deeper and the routing paths can reflect layer patterns within the class. Based on these findings, they have proposed a method to boost the robustness of the neural network.

Chen *et al.* [22] have proposed to distill knowledge from the pre-trained CNN, termed as a performer, into an explainable additive model, termed as explainer, without hurting the performance of the performer. In this case, the explainer is to paraphrase the feature representation inside of the performer by finding specific visual concepts and finally helps to understand the logic of the performer's prediction. To overcome the bias-interpreting problem, extra losses about prior weights of visual concepts have been incorporated in training.

Different from the above-mentioned methods which mainly focus on adjusting network structures, Sabour *et al.* [131] have proposed a new network named capsule nets to analyse the processing procedure. A capsule is an activity vector whose norm and orientation represent the probability and attributes respectively. To be specific, a capsule denotes a set of neurons and its activation represents a specific entity such as an object or an object part. Finally, with a dynamic routing mechanism, capsule net can decompose the entire object into a parsing tree of capsules.

Chapter 3

FAN-Face: a Simple Orthogonal Improvement to Deep Face Recognition

This chapter proposes a simple orthogonal improvement to deep face recognition. It is known that facial landmarks provide pose, expression and shape information. In addition, when matching, for example, a profile and/or expressive face to a frontal one, knowledge of these landmarks is useful for establishing correspondence which can help improve recognition. However, in prior work on face recognition, facial landmarks are only used for face cropping in order to remove scale, rotation and translation variations. We propose a simple approach to face recognition which gradually integrates features from different layers of a facial landmark localization network into different layers of the recognition network. To this end, we propose an appropriate feature integration layer which makes the features compatible before integration. We show that such a simple approach systematically improves recognition on the public face recognition datasets, setting a new state-of-the-art on LFW [59], YTF [173], IJB-B [171], IJB-C [100] and MegaFace [71] datasets under both identification and verification measurements.

The contributions of this Chapter have been published at AAAI 2020 in [191].

3.1 Introduction

Face recognition is the process of recognizing or verifying a person's identity from a given facial image or video. It is an important problem in computer vision research with many applications like access control, identity recognition in social media, and surveillance systems. With the advent of deep learning, there has been a tremendous progress in designing effective face recognition systems, yet, many applications (e.g. border control) require super-human accuracy and, as such, improving existing systems is still an active research topic. Our main contribution is a simple approach to improving deep face recognition accuracy via incorporating face-related information (e.g. pose, expression and landmark correspondence) provided by a network for facial landmark localization in order to facilitate face matching. Besides improving accuracy, our approach can be readily incorporated into all existing state-of-the-art face recognition methods.

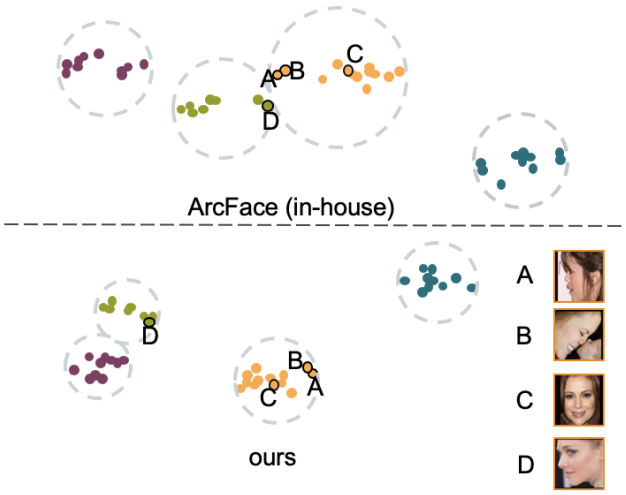


Fig. 3.1 Feature embeddings produced by our method (bottom) and our strong baseline ArcFace [34] (top). Different colours represent different identities in t-SNE space. The faces are from the CFP-FP dataset which contains frontal and profile faces. ArcFace embeddings for faces A and B are much closer to D which is from a different identity but similar pose. Overall, for all 4 identities shown the feature embeddings produced by our method are much more concentrated and separated than those of [34]. See also section 3.6.3 for a quantitative comparison between the two methods on the whole dataset where we show large improvement over [34].

The ultimate goal of face recognition is to learn a feature embedding for each face with small within-class and large between-class distances. Traditionally, this has been considered a difficult problem due to large facial appearance variations mostly caused by pose, facial expression, occlusion, illumination and age. When matching, for example, a profile A_p to a frontal A_f face of the same identity A, this distance must be smaller than the distance between A_f with another frontal face B_f of identity B. Recently, CNNs have been shown that they can learn to some extent such an embedding from large annotated face recognition datasets. Specifically, a few recent works [170, 94, 162, 34] have proposed more effective loss functions so that the learned feature embeddings for each face are both separable and discriminative.

Our work has a similar objective but departs from all the aforementioned works in that it does not propose a new loss function. In contrast, for any given loss function, in this work, we propose to learn a better feature representation for face matching and recognition by integrating, during training, features from a pre-trained network for localizing facial landmarks. A network for detecting facial landmarks is trained to learn by construction to establish correspondences between faces in any pose and facial expression independently of nuisance factors like illumination, blur, poor resolution, occlusion etc. Although it seems natural to incorporate such features in a face recognition pipeline in order to facilitate matching, to our knowledge, there is no prior work that proposes to do so. Some visual examples illustrating the enhanced discriminative properties of our method over our strong baseline [34] for the case of frontal to profile face matching are shown in Figure 3.1, while an overview of the proposed system is shown in Figure 3.2.

In summary, our **contributions** are:

- We are the first to explore how features from a pre-trained facial landmark localization network can be used to enhance face recognition accuracy. Contrary to prior pipelines for face recognition, facial landmarks are not just used for face cropping and normalization. Instead, both landmark heatmaps and features from the facial landmark network are integrated into the face recognition feature extraction process to (a) provide facial pose-, expression-, and shape-related information, and (b) help establish correspondence for improving face matching.
- We explore various architectural design choices at a network level to identify the best strategy for integration. Importantly, we propose a novel feature integration layer that is able to effectively integrate the features from the two networks although they are trained with very different objectives and loss functions.
- We conducted extensive experiments illustrating how the proposed approach, when integrated with existing state-of-the-art methods, systematically improves face recognition accuracy for a wide variety of experimental settings. Our approach sets a new state-of-the-art on the challenging LFW [59], YTF, IJB-B [171], IJB-C [100] and MegaFace [71] datasets.

The rest of this chapter is organized as follows. Section 3.2 demonstrates the overview and components of Fan-Face including heatmap integration, feature integration, the design intuitions of integration layers. Section 3.3 introduces the closely related work of the proposed method. Section 3.4 describes the implementation details, and training and evaluation datasets in experiments. Section 3.5 explores and analyses the variants under the proposed framework. Section 3.6 compares FAN-Face network with several state-of-the-art methods on 6 popular datasets. Finally, a conclusion is brought forth in Section 3.7.

3.2 Method

3.2.1 Overview

Our method is based on integration of features from two networks: a facial landmark localization network and a face recognition network. The facial landmark localization network is a pre-trained FAN [9] which has been shown to robustly detect facial landmarks (we used the 51 internal landmarks, ignoring the ones on the face boundary which are noisy due to tight cropping) across large poses, facial expressions, occlusions, illumination changes, low resolution etc., and currently represents the state-of-the-art. FAN is a stacked hourglass network [107] built using the residual block of [8]. We used two stacks as they suffice for good accuracy. The face recognition network, denoted as FRN, is a modification of ResNet [34, 51] which is the method of choice for various classification tasks (including face recognition).

The basic idea behind our method is simple: integrate features from the pre-trained FAN into FRN while training FRN. Other than that, the FRN is trained in standard ways on face

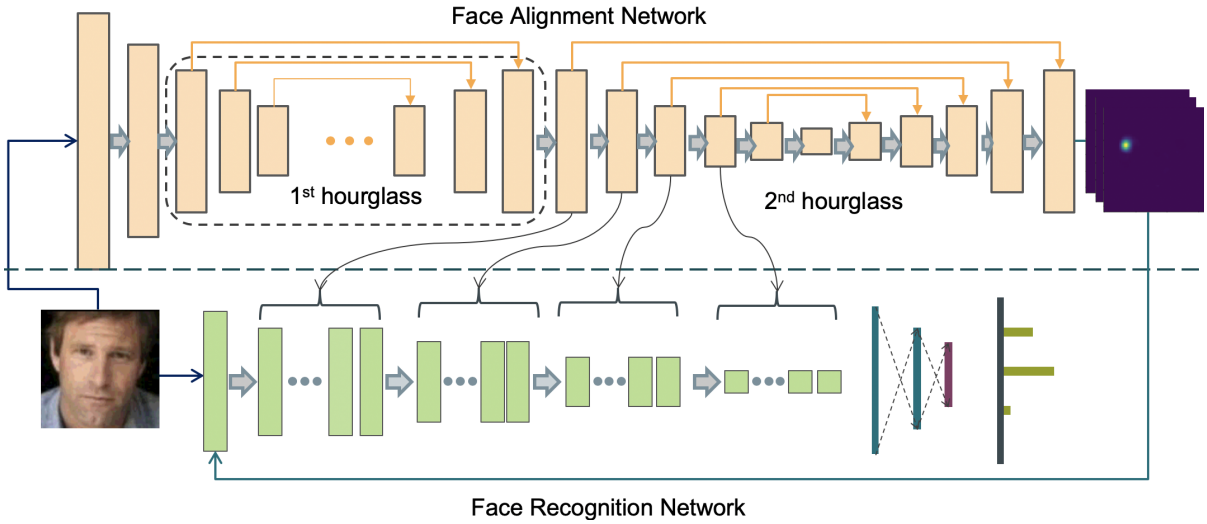


Fig. 3.2 Overview of our method: We use a pre-trained Face Alignment Network (FAN) to extract features and landmark heatmaps from the input image. The heatmaps are firstly stacked along with the image and then passed as input to a Face Recognition Network (FRN). The features (here taken from the high-to-low part of the 2-nd hourglass of FAN) are gradually integrated with features computed by FRN. Figure 3.4 shows an example of possible connectivity between the two networks. As the features from the two networks are not directly compatible, we also propose a novel feature integration layer shown in Figure 3.6.

recognition datasets. We integrate two types of features from FAN: (a) its output in the form of facial landmark heatmaps, and (b) features from different layers extracted in different resolutions. These two types of integration are detailed in the subsequent sections. An overview of our method is illustrated in Figure 3.2.

3.2.2 Heatmap Integration

Facial landmarks in FAN are localized through heatmap regression: each landmark is represented by an output channel $H_i \in R^{M_H \times M_H}, i = 1,\ldots,N$ where a 2D Gaussian is placed at the landmark's location, and then the network is trained to regress these Gaussians (known as heatmaps).

The heatmap tensor $H \in R^{N \times M_H \times M_H}$ has a number of interesting properties:

(a) it can be used to establish landmark correspondence across different face images;

(b) it captures the spatial configuration of all landmarks, and hence it captures pose, expression and shape information;

(c) as each heatmap is a confidence map, a number of works have shown that it also provides spatial context and part relationships [168].

Hence, we argue that it is natural to incorporate this tensor into FRN to facilitate face matching.

This is done as follows: each training face image $I \in R^{C \times M_I \times M_I}$ is processed by FAN which produces heatmap tensor H. The heatmaps are re-sampled to resolution $M_I \times M_I$ and then, a

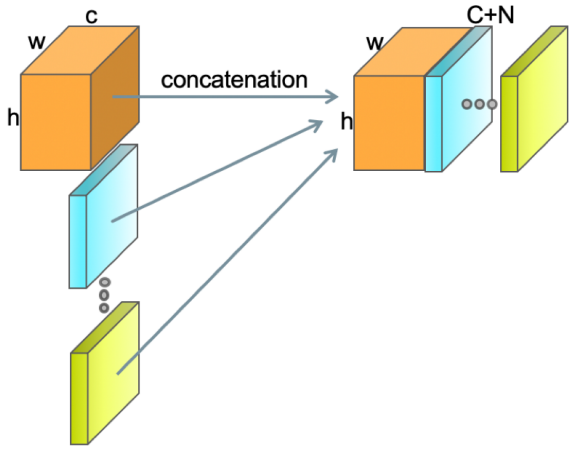


Fig. 3.3 Heatmap integration. We concat the face image with the heatmaps predicted by FAN as final input to the FRN.

stacked image-heatmap representation:

$$I_H \in R^{(C+N)\times M_I \times M_I}, \tag{3.1}$$

is used as input to train the FRN. See also Figure 3.3. Since the heatmaps capture spatial information (the x,y coordinates for each landmark can be directly obtained from $\arg\max\{H_i\}$), it is natural to directly stack them with the image. However, in Section 3.5, we also investigate whether H can be incorporated in other than the input layer of FRN. We note that image-landmark heatmap stacking as a way to guide the subsequent task has been used in a number of *low*- and *middle*-level tasks like facial part segmentation and 3D reconstruction [64]. However, to our knowledge, we are the first to investigate its usefulness for the *high*-level task of face recognition.

3.2.3 Feature Integration

The success of heatmap integration motivated us to explore whether deeper integration between FAN and FRN is possible with the goal always being to increase face recognition accuracy without significantly changing the number of the parameters of FRN. In particular, let $x_l \in R^{C_l \times H_l \times W_l}$ and $y_k \in R^{C_k \times H_k \times W_k}$ be features from the $l-$th and $k-th$ layers of FAN and FRN respectively. We choose layers l and k so that the corresponding spatial resolutions approximately match and then pass x_l through an interpolation layer so that they match completely. Following this, we compute new features $\tilde{y}_k \in R^{C_k \times H_k \times W_k}$ from:

$$\tilde{y}_k = y_k + \gamma f(x_l, y_k), \tag{3.2}$$

where f is an integration layer computing a feature combination function (to be defined below). The newly generated features $\tilde{y}_k$ are then passed to the next layer of FRN for further processing. Note that this process is applied for several layers of FRN allowing a deep integration of features from FAN to FRN. Given that the FRN is a ResNet, there is a lot of flexibility in choosing which

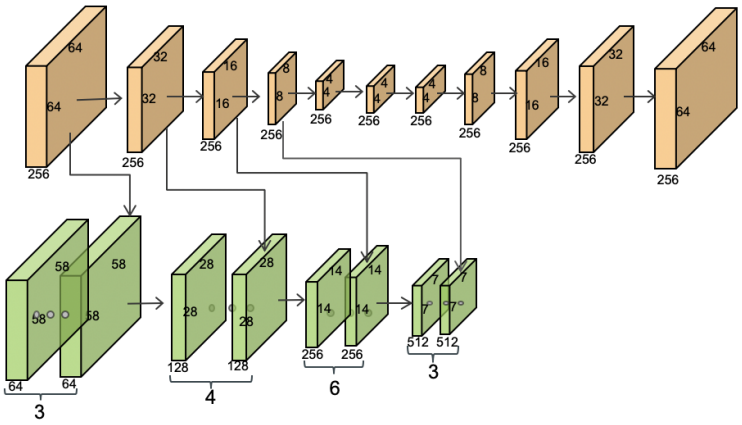


Fig. 3.4 Proposed 1:1 connectivity between FAN and FRN: at each spatial resolution, a feature tensor from the high-to-low part of the hourglass (shown in top) is combined with a feature tensor from FRN (a ResNet34 in this example shown in the bottom). The features are combined with the integration layer of Figure 3.6 and used as input to the next layer of FRN.

layers the features to be combined should be taken from, and as mentioned above, in practice, the only constraint that we apply is that the features combined have similar spatial resolutions. For examples, when the face recognition network is a ResNet-34 which has 4 stages/modules (each of them corresponding to a different spatial resolution), we always select the feature tensor at the end of each stage before down-sampling. Two instantiations of this idea are as follows.

1-1 Connectivity

In this integration scheme, we combine one feature tensor from FAN with one feature tensor from FRN for each distinct spatial resolution. Note that FAN has one top-down (high-to-low) and one bottom-up (low-to-high) part and hence for each resolution we have two parts to pick the feature tensors from. We found experimentally (see also Section 3.5) that the high-to-low part provides better features for face recognition. Figure 3.4 shows how the two networks are integrated under this scheme when FRN is a ResNet-34.

1-Many Connectivity

In this integration scheme, we combine one feature tensor from FAN with all feature tensors from FRN for each distinct spatial resolution. Also, we use learnable parameters γ_k to control the contribution of each mixing layer f, i.e. $\tilde{y}_k = y_k + \gamma_k f(x_l, y_k)$ thus allowing FRN to learn where to integrate the features from FAN. The 1-M scheme is illustrated in Figure 3.5.

3.2.4 Integration Layer

This section describes possible instantiations of the integration layer computing function f. We note that the proposed layer is able to integrate features from networks trained with very different objectives and loss functions, and hence it is one of the main contributions of this work. Our Basic Layer (denoted as BL) is depicted in Figure 3.6 and has the following main features:

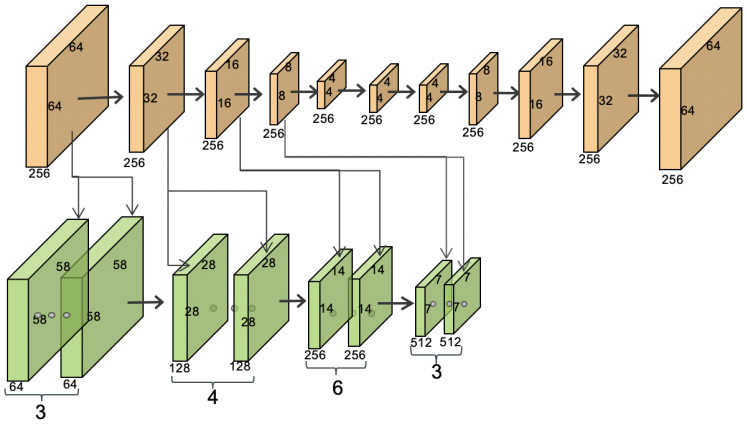


Fig. 3.5 Proposed 1:many connectivity between FAN and FRN: at each spatial resolution, a feature tensor from the high-to-low part of the hourglass (shown in top) is combined with a group of feature tensors with similar size from FRN (a ResNet34 in this example shown in bottom). The features are combined with the integration layer of Figure 3.6 and used as input to the next layer of FRN.

(a) FAN feature tensor x_l is firstly processed by a batch normalization layer that adjusts its scale (contrast). This is followed by a 1×1 convolutional layer that further aligns x_l with the FRN feature tensor y_k and makes x_l have the same number of channels as y_k.

(b) An Adaptive Instance Norm layer that makes the distribution of x_l to be similar to that of y_k. This is needed because the two feature maps are derived from different tasks: x_l pays more attention to the spatial structure of the face, while y_k focuses on identity information.

(c) Following this, the two feature tensors are combined via concatenation. Then, there is another 1×1 convolutional layer followed by batch normalization, so that the combined feature $\tilde{y}_k$ has the same number of channels as y_k so that they can be added together. The very last layer is a non-linearity in the form of PReLU.

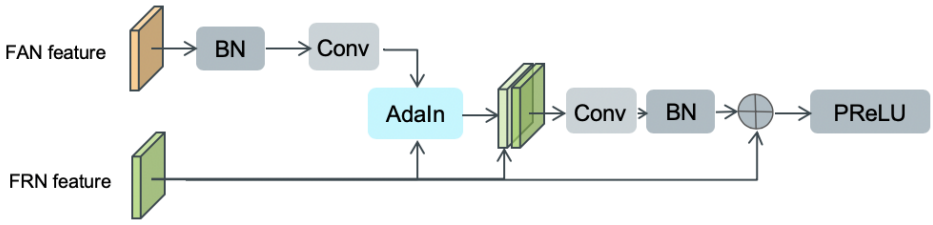


Fig. 3.6 The proposed integration layer. FAN features are processed by a batch normalization layer that adjusts their scale followed by a 1×1 conv. layer that further aligns them with FRN features. Then, An Adaptive Instance Norm makes the distribution of the two features similar. The two features are combined via concatenation. Next, there is a 1×1 conv. layer followed by a batch normalization layer so that the combined feature can be added with the input FRN feature. The very last layer is a non-linearity in the form of PReLU.

We also propose two other variants of the proposed integration layer. The Simple Layer (SL) simply adds the two feature tensors after the adaptive Instance Norm layer. The Advanced Layer

(AL) replaces the second 1×1 convolutional layer with a bottleneck layer (1×1 followed by 3×3 followed by 1×1). In Section 3.5, we compare all 3 layers, namely BL, SL, and AL.

3.3 Relationship to Previous Work

Our method is related to new loss functions brought to face recognition task. The first line of work includes a number of papers [134, 170, 94, 162, 34] which emphasize the importance of learning features that are both separable and discriminative, through the choice of a suitable loss function. Learning discriminative features is not only important for open-set recognition [94] but also for robustness across pose as naturally one of the main reasons for large within-class distances is pose variation. While the requirement for separability can be achieved with the softmax loss, learning discriminative features is more difficult as, naturally, mini-batch based training cannot capture the global feature distribution very well [170]. To this end, FaceNet [134] directly learns a mapping from face images to a compact Euclidean space in which the distance between two feature embeddings indicates the similarity of the corresponding faces such that features extracted from the same identity are as close as possible while features extracted from different identities are as far as possible. However, especially for large datasets, the number of training triplets can be prohibitively large while triplet selection also poses difficulties.

To alleviate this, more recently, the method of [170] (called Center loss) firstly realizes the importance of "center" and penalizes the Euclidean distance between the learned deep features for each face and their corresponding class centres in order to achieve intra-class concentration. The work of [94], coined SphereFace, firstly proposes to learn discriminative features in the angular domain and, to this end, it proposes to employ a multiplicative angular margin which ensures that intra-class are smaller than inter-class distances. Following the idea of working in the angular domain, the more recent works of CosFace [162] and ArcFace [34] further define concepts of "center" and "margin" to obtain highly discriminative features for face recognition.

We also notice the connection between face normalization and pose augmentation which has been shown to improve deep face recognition is through the use of models for face normalization and data augmentation. Early on, the importance of face frontalization was shown in [148]. However, frontalization for the case of large poses is a difficult problem. To handle large pose variations, the work of [99] proposes training pose-aware CNNs with data rendered by a 3D model and pose-specific frontalization. Face synthesis across pose, shape and expression for data augmentation and its effect on improving deep face recognition is systematically evaluated in [99]. Using 3DMMs for conditioning GANs for large pose frontalization is proposed in [201]. Simultaneously learning pose-invariant identity features and synthesizing faces in arbitrary poses is proposed in [154]. [32] proposes a framework for training Deep Convolutional Neural Network (DCNN) to complete the facial UV map extracted from in-the-wild images for Pose-Invariant Face Recognition. Our work also aims to extract more discriminative features, however, not via proposing a new loss function but via learning a better feature representation for recognition by integrating, features from a pre-trained facial landmark localization network. Moreover, our

approach inherently copes with large pose, not via normalization or pose augmentation but via establishing face correspondence, and typically is much more efficient than such methods for both training and testing.

3.4 Training and Implementation Details

Loss functions. To train our networks, we mostly used the ArcFace loss [34] which has been shown to outperform all other recently proposed methods like [170, 94, 162]. This is important because we show systematic improvements on top of [34] which is state-of-the-art.

Training datasets. We trained our models on 3 popular training datasets: for most of our experiments we used VGGFace2 [13] (an improved version of VGGFace [116]), containing 3.31M images of 9,131 subjects with large variations in pose, age, illumination, ethnicity and profession. This model was evaluated on IJB-B [171] and IJB-C [100] datasets. We also trained our model on MS1MV2 [34], a semi-automatically refined version of MS-Celeb-1M dataset [47] which is one of the largest wild dataset containing 98,685 celebrities and 10 million images. As an amount of noise exists in the MS-Celeb-1M dataset, the data is cleaned by [177]. There are 79,077 identities and 5 million images remaining. We also trained our model on CASIA-Webface [195] which contains 0.49M face images from 10,575 subjects. This model was evaluated on LFW [59], YTF [173], CFP-FP [136] and MegaFace [71]. In our experiments, we removed the images that belong to identities from the testing datasets.

Other hyperparameters. We followed the publicly available code of [34] for implementing and training our models. For a fair comparison, we used the same ResNet as ArcFace [34]. The block is shown in Figure 3.7, which is different from Vanilla-ResNet, and has proven to be more advanced in face recognition task in [34]. FRN and the integration layers were trained from scratch with SGD with a batch size of 512. The weight decay was set to $5e^{-4}$ and the momentum to 0.9. FAN remained fixed for the whole training procedure. All models were implemented in PyTorch [119].

Face pre-processing. We followed standard practices in face recognition [170, 94, 162, 34] to crop a face image of 112×112 (without using landmarks for alignment) which was normalized to [-1, 1]. The training faces were randomly flipped for data augmentation. The input to FAN model is up-sampled face images from 112×112 to 256×256.

3.5 Ablation Studies

In this section, we evaluate the accuracy of interesting variants and training procedures of the proposed method. The experiments are done by training the models on a randomly selected subset of 1M images from VGGFace2 dataset and evaluating them on the IJB-B dataset. FRN in all cases is a ResNet34, and unless otherwise stated all methods are trained with the ArcFace loss. All results are shown in Table 3.1. We provide results for True Acceptance Rate (TAR)

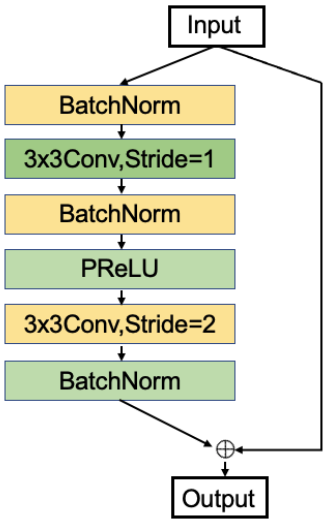


Fig. 3.7 Illustration of Residual unit in ArcFace [34]: BN-Conv-BN-PReLu-Conv-BN.

at a wide range of False Acceptance Rates (FARs), however, when we compare methods in the sections below, we primarily base our evaluations on TAR at FAR=10^{-4} as in [34].

We consider the following cases:

AS1: Heatmap integration. For all of our experiments, we used the heatmaps from the 2-nd hourglass (using the heatmaps from the 1-st hourglass gave slightly worse result due to inferior accuray in heatmap prediction). We call FAN-Face(H2), the variant of our model, obtained by simple stacking these heatmaps with FRN (see Section 3.2.2). As Table 3.1 shows, this gives some decent improvement over the baseline ArcFace with improvement from 0.859 to 0.864 at 10^{-4}. We also investigated whether stacking heatmaps with features from various layers of FRN is also beneficial. To this end, we stacked the heatmaps from second hourglass (after appropriately resizing them) along with the features produced by the last layer of FRN before each resolution drop (features from 4 different places in total, one for each spatial resolution). The performance of this variant, called FAN-Face(H2+), is shown in Table 3.1. We observe that this kind of deeper guidance using heatmaps offers good improvement for FAR=10^{-5} but no obvious improvement for FAR=10^{-4}. For FAR=10^{-5}, it increases from 0.750 to 0.771 while for FAR=10^{-4}, it only increases by 0.2%. Considering the added computation cost (in Table 3.2), we chose H2 as the better choice for the remaining setting.

AS2: Different FAN subnetworks. We are now moving to joint heatmap and feature integration. To perform feature integration, we started with 1-1 connectivity (see Section 3.2.3) and the Basic Layer (see Section 3.2.4). We call this variant FAN-Face(H2, 1-1, BL). We wanted to quantify which of the high-to-low or low-to-high subnetworks is superior. To this end, we chose the high-to-low and low-to-high subnetworks from hourglass 2. We call these variants FAN-Face(H2, 1-1, BL, h2l-2), and FAN-Face(H2, 1-1, BL, l2h-2). Table 3.1 shows the results: the high-to-low subnetwork clearly provides much better features for integration than those of low-to-high. For FAR=10^{-5}, it increases from 0.752 to 0.772 while for FAR=10^{-4}, it also gains by 0.5%. This is

	Method	10^{-5}	10^{-4}	10^{-3}	10^{-2}
Baseline	ArcFace (ResNet34)	0.747	0.859	0.929	0.966
AS1	H2	0.750	0.864	0.928	0.966
AS1	H2+	0.771	0.866	0.932	0.968
AS2	H2,1-1,BL,l2h-2	0.752	0.871	0.933	**0.970**
AS2	H2,1-1,BL,h2l-2	**0.772**	0.876	0.933	0.969
AS3	H2,1-1,AL,h2l-2	0.766	0.873	0.933	0.968
AS3	H2,1-1,SL,h2l-2	0.739	0.857	0.924	0.966
AS4	H2,1-M,BL,h2l-2	0.761	0.874	**0.934**	0.968
AS5	ResNet34-Softmax	0.619	0.786	0.898	0.964
AS5	H2,1-1,BL,h2l-2-Softmax	0.657	0.796	0.903	0.964
AS6	H2, 1-1, BL, Wing	0.769	0.870	0.932	0.966
AS6	H2, 1-1, BL, Simple	0.770	0.874	0.933	0.968
AS6	H2, 1-1, BL, HRNet	0.771	**0.877**	0.933	0.969
AS7	Multi-task[200]	0.583	0.731	0.843	0.927

Table 3.1 Verification results (%) for different variants of our method on IJB-B dataset. All models were trained on a randomly selected subset of 1M images from VGGFace2. The variants and other details are presented in Section 3.5. h2l denotes the high-to-low part in hourglass structure while l2h represents the low-to-high part in hourglass structure.

expected as low-to-high features resemble heatmaps due to appended heatmap regression loss, and we are already using heatmap information directly by stacking with the input image.

AS3: Different integration layers: Herein, we choose our best performing version so far FAN-Face(H2, 1-1, BL, h2l-2) and replace BL with the Simple and Advanced Layers, denoted as SL and AL respectively (described in Section 3.2.4). The results are shown in Table 3.1. We observe that SL performs worse because the features from FAN and from FRN are from different tasks, and directly adding them together destroys the feature semantic information. Also, there is little improvement if AL is used with more added computation cost (in Table 3.2), so we chose BL for the remaining of our experiments.

AS4: Different FAN-FRN connectivities. We compare the 1-1 with the 1-many (denoted as 1-M) connectivities (see Section 3.2.3). The results are shown in Table 3.1. Experimentally, we observe that 1-M offers no improvement over 1-1. We argue that the incurring abundant shape information on all feature maps in same stage will interfere feature learning for recognition task.

AS5: Using softmax loss. Using FAN-Face(H2, 1-1, BL, h2l-2), we also verified the improvements obtained by our method using a different loss, namely the standard softmax. See Table 3.1. For FAR=10^{-4}, FAN-Face increases the softmax baseline from 0.786 to 0.796 with 1% absolute improvement. We observe that with ArcFace loss, our method improves upon the baseline even

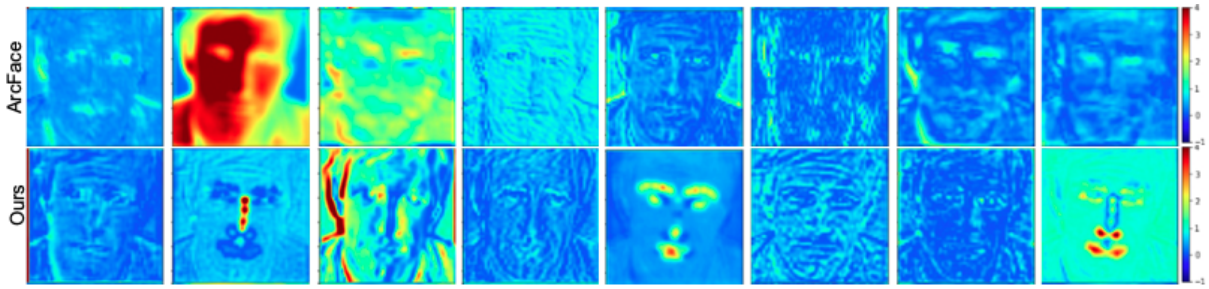


Fig. 3.8 Visualization of feature maps from ArcFace (shown in top) and our model (shown in bottom). By using FAN features to guide FRN learning, facial landmark related attention is added to the learned features.

more ($\sim 2\%$ vs 1%). This emphasizes the importance of having an advanced loss, and the complementarity of our approach with state-of-the-art losses.

AS6: Other face alignment methods. Using FAN-Face(H2, 1-1, BL), we also verified the improvements obtained by our method using different face alignment methods. AS7 shows results by replacing FAN with: a ResNet34 trained in-house to regress x,y coordinates as in [41], a ResNet34 trained in-house to regress heatmaps as in [180] and the pre-trained state-of-the-art HRNet of [142]. Results vary a bit because these networks are not equally accurate in heatmap prediction. The performance gains for Wing, Simple, HRNet are 1.1%, 1.5% and 1.8% respectively. However, overall, it is clear that the proposed feature integration strategy and integration layer are effective for all these networks, too. The above face alignment networks all have an encoder similar to h2l part in hourglass, so they can be integrated within our framework.

AS7: Multi-task [200]. A natural comparison that comes to mind is between our method and a FRN that has a second head to also predict the landmarks in a multi-task fashion, see for example [200]. For the sake of a fair comparison, we preserved the ResNet34 structure for this method and added a heatmap prediction head after layer 4 (of ResNet34). For training, the ground truth for each face is provided by the FAN network. We used L2 loss for face alignment and ArcFace loss for face recognition. The ratio between the two losses was 0.1. Table 3.1 shows the obtained results. We observe that multi-task learning does not offer competitive performance. This is mainly because of the information gap between the two tasks. The alignment task is more geometric related to low-level information while the recognition benefits from high-level semantic information.

Visualizations. Some examples of feature maps extracted in early layers produced by our model and by ArcFace are shown in Figure 3.8. While some feature maps are similar there are also a few that focus on facial landmarks. For example, the obvious landmark-related attention in the last image in the second row.

Model size and flops. The integration layers add only a very small number of extra parameters to FRN. The detailed comparison is shown in Table 3.2. Our final model adds 0.4M parameters and 0.5G flops with significant improvement in performance.

Network		Parameters(M)	Flops(G)
ArcFace [34]	ResNet34	31.8	3.71
	ResNet50	41.3	5.56
Our models with ResNet34	H2, 1-1, BL	32.2	4.25
	H2, 1-M, BL	32.7	4.75
	H2, 1-1, SL	31.9	4.17
	H2, 1-1, AL	32.9	4.87
Our models with ResNet50	H2, 1-1, BL	41.7	6.10
	H2, 1-M, BL	42.7	6.95
	H2, 1-1, SL	41.4	6.02
	H2, 1-1, AL	42.4	6.70

Table 3.2 Number of parameters and flops of different variants of our model. We compare with both ResNet34 and ResNet50 structure.

3.6 Comparison with State-of-the-Art

In this section, we compare our approach with several state-of-the-art methods on the most widely-used benchmarks for face recognition including IJB-B [171], IJB-C [100], MegaFace [71], CFP-FP [136], LFW [59], YTF [173].

Our models used: We provide the results provided by our best model FAN-Face(H2, 1-1, BL, h2l-2) which uses heatmaps from second hourglass in FAN, integrates features with 1-1 connectivity from the high-to-low subnetwork of the second hourglass, and uses as integration layer the Basic Layer of Section 3.2.4. We provide results using both ResNet34 and ResNet50 structures for FRN.

Our in-house baselines: As a very strong baseline, for all experiments, we used our implementation of ArcFace, using both ResNet34 and ResNet50. Our implementation (based on the code provided in [34]) gives slightly better results than the ones reported in the original paper. Besides, our implementation of CosFace [162] with ResNet50 was also compared because it was a strong baseline in [34].

Other methods reported: For each experiment, we also report the performance of a number of previously published methods with the results taken directly from the corresponding papers. For each method, we include, where possible, the network used (e.g. ResNet50, ResNet34, VGG) and the training set used.

3.6.1 IJB-B and IJB-C Datasets

The IJB-B dataset [171] contains 1,845 subjects (21.8K still images and 55K video frames). In total, there are 12,115 templates with 10,270 genuine matches and 8M impostor matches. The IJB-C dataset [100] is an extension of IJB-B, having 3,531 subjects (31.3K still images and

117.5K video frames). In total, there are 23,124 templates with 19,557 genuine and 15,639K impostor matches.

From the results in Tables 3.3 and 3.4, we can observe that, for both datasets, our methods provide consistently the best performance at FAR=10^{-4}, outperforming both previously proposed methods and in-house baselines, by a large margin. Specifically, on IJB-B dataset, FAN-Face(H2, 1-1, BL, h2l-2) increases $\sim 1\%$ on both ResNet34 and ResNet50 at FAR=10^{-4}. On IJBC dataset, FAN-Face(H2, 1-1, BL, h2l-2) increases $\sim 1.4\%$ and $\sim 1\%$ on ResNet34 and ResNet50 at FAR=10^{-4} respectively. Notably, our ResNet34 model provides performance which is better or in par with previously state-of-the-art ResNet50-based models with roughly 10M fewer parameters and 1.2G fewer flops.

The ROC curves are in Figure 3.9 and Figure 3.10.

Method	10^{-5}	10^{-4}	10^{-3}	10^{-2}
ResNet50 [13]	0.647	0.784	0.878	0.938
SENet50 [13]	0.671	0.800	0.888	0.949
ResNet50+SENet50 [182]	-	0.800	0.887	0.946
MNv (ResNet50) [183]	0.683	0.818	0.902	0.955
MNvc (ResNet50) [183]	0.708	0.831	0.909	0.958
ResNet50+DCN(Kpts) [182]	-	0.850	0.927	0.970
ResNet50+DCN(Divs) [182]	-	0.841	0.930	0.972
SENet50+DCN(Kpts) [182]	-	0.846	0.935	0.974
SENet50+DCN(Divs) [182]	-	0.849	0.937	0.975
ArcFace (ResNet50) [34]	0.812	0.898	0.944	**0.976**
ArcFace (ResNet34, in-house)	0.775	0.885	0.940	0.973
Ours (ResNet34)	0.817	0.900	0.945	0.974
CosFace (ResNet50, in-house)	0.806	0.895	0.945	0.972
ArcFace (ResNet50, in-house)	0.814	0.902	0.946	0.974
Ours (ResNet50)	**0.835**	**0.911**	**0.947**	0.975

Table 3.3 Evaluation of different methods for 1:1 verification on IJB-B dataset. All methods were trained on VGGFace2 dataset.

We also experiment on conducted the MS1MV2 [34] using ResNet100. Models are trained from scratch with an SGD optimizer, momentum of 0.9, weight decay of $5e-4$ and batch size of 512. The initial learning rate is 0.1 and reduced by a factor of 10 at epoch 10, 18, 22 with total 24 epochs. Results are shown in Table 3.5. We observe that our proposed method outperforms ArcFace on verification by $\sim 1\%$, rank 1 identification by $\sim 1\%$. We attribute this to the facial structure prior from alignment networks.

Method	10^{-5}	10^{-4}	10^{-3}	10^{-2}
ResNet50 [13]	0.734	0.825	0.900	0.950
SENet50 [13]	0.747	0.840	0.910	0.960
ResNet50+SENet50 [182]	-	0.841	0.909	0.957
MNv (ResNet50) [183]	0.755	0.852	0.920	0.965
MNvc (ResNet50) [183]	0.771	0.862	0.927	0.968
ResNet50+DCN(Kpts) [182]	-	0.867	0.940	0.979
ResNet50+DCN(Divs) [182]	-	0.880	0.944	0.981
SENet50+DCN(Kpts) [182]	-	0.874	0.944	0.981
SENet50+DCN(Divs) [182]	-	0.885	0.947	**0.983**
ArcFace(ResNet50) [34]	0.861	0.921	0.959	0.982
ArcFace (ResNet34, in-house)	0.851	0.912	0.955	0.981
Ours(ResNet34)	0.873	0.926	**0.960**	0.981
CosFace (ResNet50, in-house)	0.864	0.918	0.952	0.980
ArcFace (ResNet50, in-house)	0.872	0.924	0.959	0.982
Ours(ResNet50)	**0.874**	**0.935**	0.959	0.982

Table 3.4 Evaluation of different methods for 1:1 verification on IJB-C dataset. All methods were trained on VGGFace2 dataset.

Methods/Datasets	IJB-B			IJB-C		
	Ver (%)	Id (%)		Ver (%)	Id (%)	
	10^{-4}	rank=1	rank=5	10^{-4}	rank=1	rank=5
ArcFace (R100, in-house)	94.5	93.2	95.8	95.9	94.4	96.5
Ours(R100)	**95.4**	**93.7**	**96.1**	**96.8**	**94.8**	**97.0**

Table 3.5 Results (%) of our method and ArcFace (in-house) on MS1MV2 using ResNet-100. Verification (Ver) is at FAR=10^{-4}. Identification (Id) is using gallery 2.

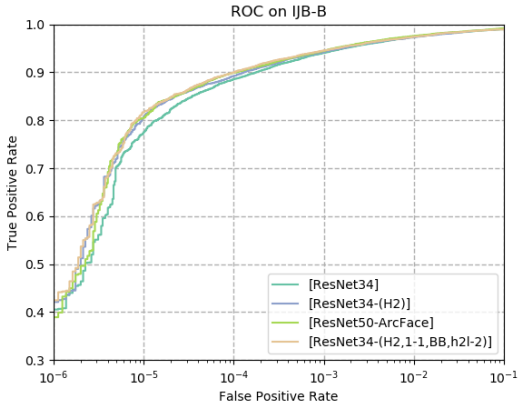


Fig. 3.9 ROC curves of 1:1 verification protocol on the IJB-B

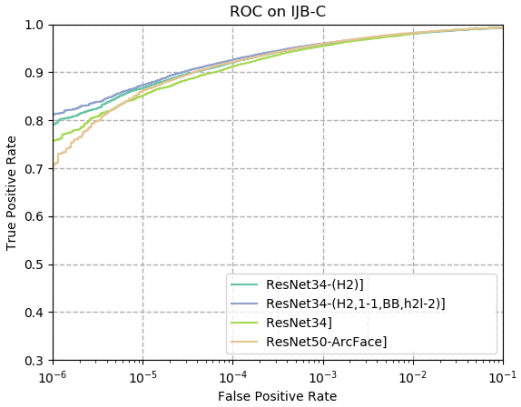


Fig. 3.10 ROC curves of 1:1 verification protocol on the IJB-C

3.6.2 MegaFace Dataset

MegaFace [71] is a dataset for evaluating the performance of face recognition at million-level distractors. It includes 1M images of 690K different individuals as the gallery set and 100K photos of 530 unique individuals from FaceScrub [108] as the probe set. In MegaFace, there are two testing scenarios, identification and verification. Here, and for the sake of fair comparison with prior work, all the proposed and in-house models were trained on the CASIA dataset [195]. Models are trained from scratch with an SGD optimizer, momentum of 0.9, weight decay of $5e-4$ and batch size of 512. The initial learning rate is 0.1 and reduced by a factor of 10 at epoch 20, 29, 37, 46 with total 50 epochs.

Method	Network	Identification (%)	Verification (%)
Softmax [94]	ResNet64	54.86	65.93
Center Loss [170]	modified ResNet	65.49	80.14
SphereFace [94]	ResNet64	72.73	85.56
CosFace [162]	ResNet64	77.11	89.88
ArcFace [34]	ResNet50	77.50	92.34
ArcFace (in-house)	ResNet34	75.52	89.53
Ours	ResNet-34	77.54	92.06
CosFace (in-house)	ResNet50	75.93	91.02
ArcFace (in-house)	ResNet50	76.44	91.44
Ours	ResNet-50	**78.32**	**92.83**

Table 3.6 Identification and verification results on MegaFace Challenge 1. Identification refers to rank-1 face identification accuracy and Verification refers to face verification TAR (True Acceptance Rate) at 10^{-6} FAR (False Acceptance Rate). All methods were trained on CASIA dataset.

Table 3.6 shows the obtained results. We observe that our ResNet50 model outperforms all previous methods and in-house baselines, significantly. Again, our ResNet34 model is the second best performing model, slightly outperforming the original implementation of ArcFace [34] for identification which however used a ResNet50 model. To be specific, our proposed method outperforms Arcface (in-house) with $\sim 2\%$ on identification task, and $\sim 1.5\%$ on verification task with ResNet34 network structure. Besides, our proposed method outperforms. Similar finding can be seen in ResNet50 structure. Although the in-house models are inferior to the public results due to the unaligned-cropped input, our proposed methods narrow the gap and even achieve better performance. The superior performance of proposed methods verifies the effectiveness and efficiency of using information from alignment model.

3.6.3 CFP-FP Dataset

As in Figure 3.1 provided only a few examples illustrating that our method is more effective than our strong baseline (ArcFace) for computing features that are more discriminative across large poses, herein, we evaluate both models quantitatively on the whole Celebrities in Frontal-Profile (CFP) [136] dataset. CFP is a challenging dataset to evaluate the performance of models for frontal to profile face verification in-the-wild. It has 500 celebrities, with 10 frontal and 4 profile face images. We evaluated our models on the Frontal-Profile setting(CFP-FP) and report the results in Table 3.7. As we may observe, our model provides obvious improvement over our implementation of ArcFace for both ResNet34 and ResNet50.

Method	CFP-FP(%)
ArcFace(CASIA,R50) [34]	95.56
ArcFace (ResNet34,in-house)	94.63
ArcFace (ResNet50,in-house)	95.49
Ours (ResNet34)	95.46
Ours (ResNet50)	**95.87**

Table 3.7 Verification Results(%) on CFP-FP. All models were trained on CASIA.

3.6.4 LFW and YTF Datasets

We also report results on LFW [59] (13,233 web-collected images from 5,749 individuals) and YTF [173] (3,425 videos from 1,595 different identities). As typical in literature, our models and in-house baselines, were trained on CASIA dataset [195]. Following the unrestricted protocol with labelled outside data [59], we report the performance our models on 6,000 face pairs from LFW and 5,000 videos pairs from YTF in Table 3.8. As in our previous experiments, our ResNet50 model performs the best while our ResNet34 is the second best along with the ArcFace ResNet50-based model of [34].

Method	Network	LFW (%)	YTF (%)
Softmax [94]	ResNet64	97.88	93.1
HiReST-9 [175]	AlexNet [77]	99.03	95.4
SphereFace [94]	ResNet64	99.42	95.0
CosFace [162]	ResNet64	99.33	96.1
ArcFace [34]	ResNet50	99.53	-
ArcFace (in-house)	ResNet34	99.40	95.42
Ours	ResNet-34	99.52	96.34
CosFace (in-house)	ResNet50	99.30	95.78
ArcFace (in-house)	ResNet50	99.47	96.13
Ours	ResNet50	**99.56**	**96.72**

Table 3.8 Verification performance (%) on LFW and YTF datasets.

3.7 Conclusion

We proposed a system that uses features from a pre-trained facial landmark localization network to enhance face recognition accuracy. In our system, both landmark heatmaps and features from the facial landmark network are integrated into the face recognition feature extraction process to provide face-related information and establish correspondence for face matching because landmarks produced by landmark detection methods can be used to access the same semantic features (like the tip of the nose) between different faces. We explored various architectural design choices at a network level to identify the best strategy for integration and, we proposed a novel feature integration layer that is able to effectively integrate the features from the two networks. We conducted extensive experiments illustrating how the proposed approach, when integrated with existing state-of-the-art methods, systematically improves face recognition accuracy for a wide variety of experimental settings. Our approach is also shown to produce state-of-the-art results on the challenging IJB-B, IJB-C and MegaFace datasets.

Chapter 4

Knowledge distillation via softmax regression representation learning

This chapter addresses the problem of model compression via knowledge distillation. Generally, knowledge distillation tries to learn a low-capacity student to mimic the behaviour of high-capacity teacher and finally improve the student network's performance. Normally, the current knowledge distillation methods can be categorized into knowledge transfer-based methods and feature relationship transfer-based methods. Here, we advocate for a method that optimizes the output feature of the penultimate layer of the student network and hence is directly related to representation learning. To this end, we firstly propose a direct feature matching approach which focuses on optimizing the student's penultimate layer only. Secondly and more importantly, because feature matching does not take into account the classification problem at hand, we propose a second approach that decouples representation learning and classification. We present a method to utilize the teacher's pre-trained classifier to train the student's penultimate layer feature. In particular, for the same input image, we wish the teacher's and student's feature to produce the same output when passed through the teacher's classifier, which is achieved with a simple L_2 loss. Our method is extremely simple to implement and straightforward to train and is shown to consistently outperform previous state-of-the-art methods over a large set of experimental settings including different (a) network architectures, (b) teacher-student capacities, (c) datasets, and (d) tasks (object classification, face alignment and face recognition), (e) domains. Besides, we also extend our method to learning from extra unlabeled data with teacher trained on labeled datasets only. For instance, we use VGGFace2 as unlabeled dataset to improve the student's performance trained on UMDFace. The experimental results show that extra introduced data can further boost the student's performance. These results validate the newly proposed method's superiority and generality in both object classification and face recognition tasks.

The contributions of this Chapter have been published at ICLR2021 [194].

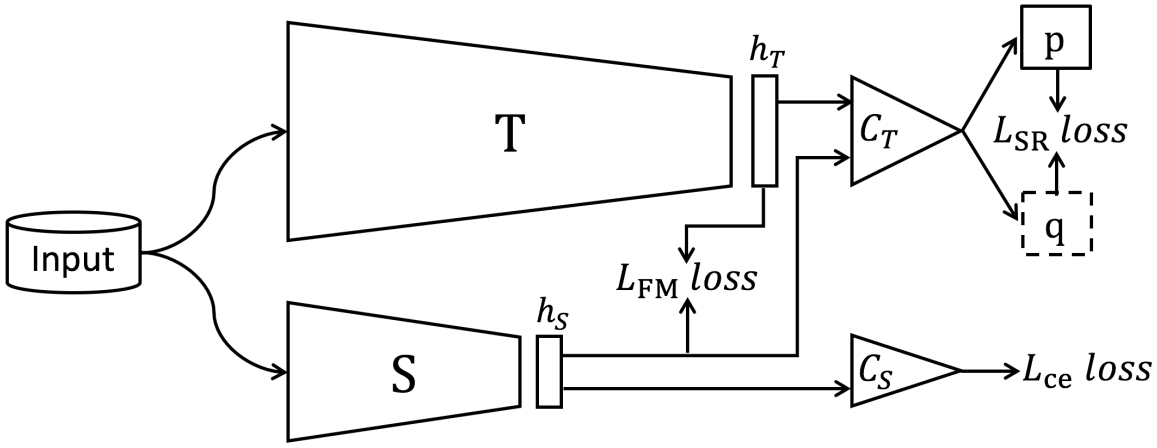


Fig. 4.1 Our method performs knowledge distillation by minimizing the discrepancy between the penultimate feature representations h_T and h_S of the teacher and the student, respectively. To this end, we propose to use two losses: (a) the Feature Matching loss L_{FM}, and (b) the so-called Softmax Regression loss L_{SR}. In contrary to L_{FM}, our main contribution, L_{SR}, is designed to take into account the classification task at hand. To this end, L_{SR} imposes that for the same input image, the teacher's and student's feature produce the same output when passed through the teacher's *pre-trained and frozen* classifier. Note that, for simplicity, the function for making the feature dimensionality of h_T and h_S the same is not shown.

4.1 Introduction

Recently, there has been a great amount of research effort to make deep model lightweight so that they can be deployed in devices with limited resources. To this end, several approaches for model compression have been proposed, including network pruning [48, 80], network quantization [125, 174], knowledge transfer/distillation [55, 204], and neural architecture search [229, 90]. Knowledge distillation [7, 55] aims to transfer knowledge from one network (the so-called "teacher") to another (the so-called "student"). Typically, the teacher is a high-capacity model capable of achieving high accuracy, while the student is a compact model with many fewer parameters, thus also requiring much less computation. The goal of knowledge distillation is to use the teacher to improve the training of the student and push its accuracy closer to that of the teacher.

The rationale behind knowledge distillation can be explained from an optimization perspective: there is evidence that high capacity models (i.e. the teacher) can find good local minima due to over-parameterization [39, 140]. In knowledge distillation, such models are used to facilitate the optimization of lower capacity models (i.e. the student) during training. For example, in the seminal work of [55], the softmax outputs of the teacher provide extra supervisory signals of inter-class similarities which facilitate the training of the student. In other influential works, intermediate representations extracted from the teacher such as feature tensors [129] or attention maps [204] have been used to define auxiliary loss functions used in the optimization of the student. Training a network whose output feature representation is rich and powerful has been shown crucial for achieving high accuracy for the subsequent classification task in recent works

in both unsupervised and supervised learning, see for example [24, 49] and [69]. Hence, in this thesis, we are advocating for representation learning-based knowledge distillation by optimizing the student's penultimate layer output feature. If we are able to do this effectively, we expect (and show experimentally) to end up with a student network which can generalize better than one trained with logit matching as in the vanilla KD paper of [55]. However, accomplishing this task with direct feature matching (between the student and the teacher) as in [129, 204] is difficult due to the lower representation capacity of the student and often leads to sub-optimal results compared to KD. Furthermore, this approach imposes a strict bit-by-bit match which does not take into account the classification problem at hand.

Main contributions: To accomplish the aforementioned goal we propose two loss functions: *The first loss function*, akin to [129, 204], is based on direct feature matching but focuses on optimizing the student's penultimate layer feature only. Because direct feature matching might be difficult due to the lower representation capacity of the student and, more importantly, is detached from the classification task at hand, we also propose *a second loss function*: we propose to decouple representation learning and classification and utilize the teacher's pre-trained classifier to train the student's penultimate layer feature. In particular, for the same input image, we wish the teacher's and student's feature to produce the same output when passed through the *teacher's* classifier, which is achieved with a simple L_2 loss (see Figure 4.1). This softmax regression projection is used to retain from the student's feature the information that is relevant to classification, but since the projection matrix is pre-trained (learned during the teacher's training phase) this does not compromise the representational power of the student's feature. Besides, our proposed method is easy to incorporate to the distillation on unlabeled data, in which teacher's predictions on both labeled and unlabeled dataset are taken to improve student's performance, and experimentally verifies that our proposed method is able to maintain its superior performance to others.

Main results: Our method has two advantages: (1) It is simple and straightforward to implement. (2) It consistently outperforms state-of-the-art methods over a large set of experimental settings including different (a) network architectures (WideResNets, ResNets, MobileNets), (b) teacher-student capacities, (c) datasets (CIFAR10/100, ImageNet, MegaFace), (d) tasks (image classification, face recognition, face alignment) (e) domains (real-to-binary).

The rest of this chapter is organized as follows. Section 4.2.1 describes the proposed knowledge distillation method called Softmax Regression Representation Learning and also its extension to unlabeled data. Section 4.3 introduces the closely related work and our differences from them. Section 4.4 investigates and explores the variants under the proposed framework, and then compare with state-of-the-art methods across different datasets and tasks. Section 4.5 studies how to improve current performance on general object classification (CIFAR100, ImageNet-1K) and Face recognition with extra unlabeled data. Finally, a conclusion is brought forth in Section 4.6.

4.2 Method

4.2.1 Softmax Regression Representation Learning

We denote by T and S the teacher and student networks respectively. We split these networks into two parts:

1. A convolutional feature extractor $f_{Net}, Net = \{T, S\}$, the output of which at the i-th layer is a feature tensor $F^i_{Net} \in \mathbb{R}^{C^i_{Net} \times H^i \times W^i}$, where C^i_{Net} is the output feature dimensionality, and H^i, W^i represent the output spatial dimensions. We also denote by $h_{Net} = \sum_{h=1}^{H^L} \sum_{w=1}^{W^L} F^L_{Net} \in \mathbb{R}^{C^L_{Net}}$ the last layer feature representation learned by f_{Net}.

2. A projection matrix $W_{Net} \in \mathbb{R}^{C^L_{Net} \times K}$ which projects the feature representation h_{Net} into K class logits $z^i_{Net}, i = 1, \ldots, K$, followed by the softmax function $s(z^i_{Net}) = \frac{\exp(z^i_{Net}/\tau)}{\sum_j \exp(z^j_{Net}/\tau)}$ with temperature τ ($\tau = 1$ for Cross Entropy loss) which put together form a softmax regression classifier into K classes.

Traditional Knowledge Distillation (KD) [55] trains the student with the following loss:

$$L_{KD} = -\sum_{k=1}^{K} s(z^k_T) \log s(z^k_S), \tag{4.1}$$

so that the discrepancy between the teacher's and student's classifiers is directly minimized.

FitNet [129] matches intermediate feature representations. For the i-th layer, the following loss is defined:

$$L_{Fit} = \left\| F^i_T - r(F^i_S) \right\|^2, \tag{4.2}$$

where $r(.)$ is a function for matching the feature tensor dimensions.

In our work, we propose to minimize the discrepancy between the representations h_T and h_S. To accomplish this goal, we propose to use two losses. The first one is an L_2 feature matching loss:

$$L_{FM} = \|h_T - h_S\|^2, \tag{4.3}$$

where for notational simplicity we dropped the dependency on $r(.)$. Hence, L_{FM} loss is a simplified FitNet loss which focuses only on the final representation learned. The intuition for this is that this feature is directly connected to the classifier and hence imposing the student's feature to be similar to that of the teacher could have more impact on classification accuracy. Moreover, it might be questionable why one should optimize for other intermediate representations as in [129] especially when the student is a network of lower representational capacity. In Section 4.4.1: *Where should the losses be applied?*, we confirm that L_{FM} alone has a positive impact but feature matching in other layers is not helpful.

We found L_{FM} to be effective but only to limited extent. One disadvantage of L_{FM} and, in general, of all feature matching losses e.g. [129, 204], is that it treats each channel dimension in the feature space independently, and ignores the inter-channel dependencies of the feature

representations h_S and h_T for the final classification. This is in contrast to the original logit matching loss proposed by Hinton *et al.* in [55] which directly targets classification accuracy. To alleviate the aforementioned problem, in this work, we propose a second loss for optimizing h_S which is directly linked with classification accuracy. To this end, we will use the teacher's *pre-trained* Softmax Regression (SR) classifier.

Let us denote by p the output of the teacher network when fed with some input image x. Let us also feed the same image through the student network to obtain feature $h_S(x)$. Finally, let us pass $h_S(x)$ through the teacher's SR classifier to obtain output q. See also Fig. 4.1. Our loss is defined as:

$$L_{SR} = -p \log q. \tag{4.4}$$

At this point, we make the following two observations: (1) If $p = q$ (and since the teacher's classifier is frozen), then this implies that $h_S(x) = h_T(x)$ which shows that indeed Eq. (4.4) optimizes the student's feature representation h_S (h_T is also frozen). (2) The loss of Eq.(4.4) can be written as:

$$L_{SR} = -s(W_T' h_T) \log s(W_T' h_S). \tag{4.5}$$

Now let us now write KD loss in a similar way:

$$L_{KD} = -s(W_T' h_T) \log s(W_s' h_S). \tag{4.6}$$

By comparing Eq. (4.5) with Eq. (4.6), we see that the only difference in our method is that the frozen, pre-trained teacher's classifier is used for both teacher and the student. On the contrary, in KD, W_S is also optimized. This gives more degrees of freedom to the optimization algorithm, in particular, to adjust the weights of *both* the student's feature extractor f_S *and* the student's classifier W_S in order to minimize the loss. This has an impact on the learning of the student's feature representation h_S which, in turn, hinders the generalization capability of the student on the test set. We confirm this hypothesis with the experiment of Section 4.4.1: *Transferability of representations*.

Finally, we note that we found that, in practice, an L_2 loss between the logits:

$$L_{SR} = \left\| W_T' h_T - W_T' h_S \right\|^2 = \| h_T - h_S \|^2_{W_T} \tag{4.7}$$

works slightly better than the cross-entropy loss. The comparison between different types of losses for L_{SR} is given in the experiment section.

Overall, in our method, we train the student network using three losses:

$$L = L_{CE} + \alpha L_{FM} + \beta L_{SR}, \tag{4.8}$$

where α and β are the weights used to scale the losses. The teacher network is pretrained and fixed during training the student. L_{CE} is the standard loss based on ground truth labels for the

task in hand (e.g. cross-entropy loss for image classification). Note that this results in a very simple algorithm for training the student summarized in Algorithm 1.

Algorithm 1 Knowledge distillation via Softmax Regression Representation Learning

Input: Teacher network T, Student network S, input image x, ground truth label y, weights α, β.

1. Input x to S to obtain feature h_S and class prediction $\hat{y}$. Calculate cross entropy loss $L_{CE}(\hat{y}, y)$;
2. Input x to T to obtain feature h_T. Calculate distillation losses from Eqs. (4.3) and Eqs. (4.7);
3. Update S by optimizing Eq. (4.8)

Output: the updated S.

4.2.2 Extension to Unlabeled Data

Conventional knowledge distillation typically considers a setting in which teacher and student are trained on the same labeled datasets. On the contrary, the typical setting in semi-supervised learning is to consider a large amount of unlabeled data together with the labeled training set. The unlabeled data is then exploited together with the labeled data to improve the performance on the original supervised task. Motivated by the success of semi-supervised learning, we propose to investigate out-of-distribution unlabeled data to further boost the effectiveness of the transfer from the teacher to student network. In this section, we formally describe the problem setting. We further introduce two common techniques commonly used in semi-supervised learning with the aim of validating whether their effectiveness is also translated into our setting.

Given training data consisting of (1) labeled training images $\mathtt{D} = \{(x_i, y_i)\}_{i=1}^{N_l}$ and (2) unlabeled images $\mathtt{U} = \{x_i\}_{i=1}^{N_u}$, our goal is to improve student's performance on the same test dataset by training it on $\mathtt{D} \cup \mathtt{U}$. The general loss function for learning the student is expressed as:

$$L_{total} = L_{CE}(\mathtt{D}) + \gamma L_{KD}(\mathtt{D}) + \lambda L_{KD}(\mathtt{U}) \tag{4.9}$$

where L_{CE} is a standard cross-entropy classification loss from $\mathtt{D}$, L_{KD} is a knowledge distillation loss (we discuss its particular implementations like KD [55], AT [204] in experiment), γ and λ are non-negative scalar weights to balance losses from the labeled and unlabeled datasets.

The rationale behind the knowledge distillation loss on unlabeled data is that the teacher network trained on labeled datasets is capable of capturing rich relational information among labels both in class space and feature space. For instance, in face recognition task, the training set and testing set have no overlap in categories but the model trained on the training set with classification loss is able to do verification on unseen verification set with extracted feature embedding. Thus, we argue that for an input which belongs to the same modality as the labeled dataset or drawn from a related, but not necessarily the same distribution, the teacher network is able to model its multi-modal distribution of visual concepts within the labeled dataset space

and finally guide the student to learn a better visual representation and classifier. We extend this hypothesis in both class and feature space by instantiating this idea with state-of-the-art knowledge distillation methods and then verify its effectiveness by conducting experiments on both object classification and face recognition tasks.

For KD [55] and AT [204], we only need to instantiate L_{KD} with KL divergence loss at teacher's output and L_2 loss on inner feature tensors. In particular, For KD on unlabeled datasets $\mathtt{D}$, it works like pseudo labeling, teacher network provides pseudo labels $\hat{y}$ for $x \in \mathtt{U}$ and thus student will learn from $\mathtt{U} = (x_i, \hat{y}_i)_{i=1}^{N_u}$. However, this idea is not so straightforward to be applied to [153], where the knowledge is defined as normalized feature embedding, and the contrastive loss is to capture correlations and higher-order output dependencies. In practice, for features extracted prior to the classifier C_T and C_S, a linear projection is introduced to transform them to same dimension as $F_T(x)$ and $F_S(x)$.

$$L_{KD} = -log[h(F_T(x), F_S(x))] - \sum_{\bar{x} \in D_N} log[1 - (h(F_T(x), F_S(\bar{x}))] \tag{4.10}$$

For the labeled data, $\mathtt{D}_N$ is a set of negative samples that are drawn at random from $\mathtt{D}$ except images with the same label as x. The extension of [153] to unlabeled data is different from labeled data as there is no label provided in negative sample selection. To solve this issue, we follow self-supervised learning [179] to randomly select other images as negative samples. To this end, the whole contrastive representation distillation loss is made up of the supervised contrastive loss trained on $\mathtt{D}$ and the self-supervised contrastive loss trained on $\mathtt{U}$.

MixUp Labeled and Unlabeled Data

We follow mixup [209] to "mix" the data where each labeled sample is interpolated with another sample chosen from the mini-batch of unlabeled datasets. Specifically, for a pair of samples $(x_l \in \mathtt{D}, x_u \in \mathtt{U})$ and their predictions from teacher network (p_l, p_u), the mixed (x', y') is computed by:

$$\begin{aligned} \lambda &\sim Beta(\alpha, \alpha) \\ \lambda' &= max(\lambda, 1-\lambda) \\ x' &= \lambda' x_l + (1-\lambda') x_u \\ y' &= \lambda' p_l + (1-\lambda') p_u \end{aligned} \tag{4.11}$$

Finally, the student network with input x' is trained to minimize the KL divergence with y':

$$L_{mixup} = \tau^2 KL(S(x')/\tau || y'/\tau) \tag{4.12}$$

Hence, the student network is expected to generalize well to the evaluation data by enforcing it to imitate the teacher model on the mixup images.

The algorithms for vanilla knowledge distillation on both labeled and unlabeled datasets are summarized in Algorithm 2.

Algorithm 2 Pseudocode of KD extension to unlabeled data in a PyTorch-like style.

```
# T: teacher network, S: student network
# L_loader: labeled dataloader, U_loader: unlabeled dataloader
T.eval()
S.train()
for i, (x_l,y_l,x_u) in zip(L_loader,U_loader):# load a minibatch x with N samples
   # KD loss on labeled data
   pred_t_l = T(x_l)
   pred_s_l = S(x_l)
   loss_l = L_KD(pred_s_l, pred_t_l.detach())
   # cross-entropy loss on labeled data
   loss_ce = CrossEntropyLoss(pred_s_l, y_l)
   # KD loss on labeled data
   pred_t_u = T(x_u)
   pred_s_u = S(x_u)
   loss_u = L_KD(pred_s_u, pred_t_u.detach())
   # mix up labeled and unlabled data
   lam = numpy.random.beta(alpha, alpha)
   index = torch.randperm(x_l.size(0))
   x_mix = lam * x_l + (1. - lam) * x_u[index]
   y_mix = lam * pred_s_l + (1. - lam) * pred_t_u[index]
   pred_s_mix = S(x_mix)
   loss_mixup = L_KD(pred_s_mix, y_mix.detach())
   optimizer.zero_grad()
   {loss_ce,loss_l,loss_u,loss_mixup}.backward()
   optimizer.step()
```

4.3 Relationship to Previous Work

Our method is related to metric learning for learning discriminative feature embedding. The core of metric learning [184, 169] is to project data into a feature embedding space where samples from the same class are close together while samples from the different classes are far apart. Successful metric learning has achieved competitive performance in computer vision tasks like face recognition [134], person re-identification [181], and fine-grained retrieval. In these tasks, feature representation is used to determine the similarity of given input pairs, while the subsequent linear classifier is discarded. Feature learning in knowledge distillation can be viewed as a certain type of metric learning, which targets at driving student learning with prior distribution in embedding space provided by the teacher network that has superior ability over student in capturing discriminative feature. This characteristic is especially shared in feature relationship transferring knowledge distillation. These works always select the distillation position at the later part of the network because early layers capture features of fine-scale while later layers focus more on coarse-scale abstract feature. Our method decouples student learning into feature representation learning and classification. The feature representation learning parts takes a different view by using teacher's linear classifier as a guidance enforcing student to directly mimic teacher's discriminative properties.

We also notice the connection between our approach and Semi-supervised learning (SSL). SSL is designed to use a large amount of unlabeled data to help supervised learning on small amount of labeled data. Self-training (also called pseudo-labeling [81]) [128, 68, 2] is a popular solution in SSL. The conventional steps of self-training are unfolded as follows: Firstly, it trains a good teacher model on labeled data. Then, it uses the teacher to provide pseudo labels for unlabeled data. Finally, it combines teacher's predictions on unlabeled data together with annotations on labeled data to train the student network. Another line of SSL is based on consistency regularization. The key point is to encourage the model to produce the same output distribution when its inputs are perturbed [132, 79, 101]. In practice, consistency regularization

is to enforce a classifier to output the same class distribution when fed into augmentations of the same input sample. The rationale behind this idea is the smoothness assumption in classification boundary. Knowledge distillation on unlabeled data shares some similarities with SSL because both tasks are provided with labeled and unlabeled samples in experiment setting. The difference comes from the data distribution. In SSL, both labeled and unlabeled datasets share the same class set. For example, SSL often splits the whole annotated dataset into two splits: data with ground-truth labels in training and data without ground-truth labels in training. However, in our unlabeled knowledge distillation setting, the unlabeled data is from another distribution with different classes, and even with the same label, unlabeled datasets have a clear domain gap with the labeled one.

Our approach shares the same idea with mixup [209], a regularisation technique which encourages deep neural networks to behave linearly between pairs of data points and has been utilized successfully in supervised learning [209, 157] and semi-supervised learning [158, 6]. Unlike current mixup-based methods, we apply mixup to effectively leverage labeled data of known classes for unlabeled classes. Note that, existing mixup-based methods assume that every class of interest has clean label. However, in our setting, the pseudo label for unlabeled data is from teacher's prediction and may not belong to defined classes.

4.4 Experiments on Labeled Datasets

In this section, we first investigate the components of our method by analysing variants. Then we compare the proposed method with existing knowledge distillation methods across different tasks, network structures, and domains.

Group Name	Output size	WRN-D-k	WRN-16-4
conv1	32x32	3x3,16	
conv2	32x32	$\begin{bmatrix} 3*3, 16k \\ 3*3, 16k \end{bmatrix}$ xd	$\begin{bmatrix} 3*3, 64 \\ 3*3, 64 \end{bmatrix}$ x2
conv3	16x16	$\begin{bmatrix} 3*3, 32k \\ 3*3, 32k \end{bmatrix}$ xd	$\begin{bmatrix} 3*3, 128 \\ 3*3, 128 \end{bmatrix}$ x2
conv4	8x8	$\begin{bmatrix} 3*3, 64k \\ 3*3, 64k \end{bmatrix}$ xd	$\begin{bmatrix} 3*3, 256 \\ 3*3, 256 \end{bmatrix}$ x2
	1x1	average pool, 100-c fc, softmax	

Table 4.1 Structure of the Wide ResNet (WRN) networks used in our experiments. c denotes number of classes. For CIFAR10, $c = 10$. For CIFAR100, $c = 100$. $d = (D-4)/6$.

4.4.1 Ablation Studies

We conducted a set of ablation studies on CIFAR100 (see Section 4.4.2) using a Wide ResNet (WRN) for both teacher (WRN-40-4) and student (WRN-16-4). The architecture is described in Table 4.1.

Are both L_{FM} and L_{SR} useful? To answer this question, we ran three experiments: using L_{FM} alone, L_{SR} alone, and combining them together $L_{FM}+L_{SR}$. The results of Table 4.2 (first 3 rows) clearly show that all proposed variants offer significant gains: when using L_{FM} and L_{SR} alone, $\sim 1\%$ and $\sim 2\%$ improvements in Top-1 accuracy were obtained. Moreover, when combining them together, an additional $\sim 0.4\%$ improvement was gained. Importantly, the results show that L_{SR} is significantly more effective than L_{FM}. We further note at this point that we found that L_{FM} offers diminishing gains on ImageNet-1K experiments.

Method	Layer	Top-1 (%)	Top-5 (%)
Student (WRN-16-4)		76.97	93.89
Teacher (WRN-40-4)		79.50	94.57
L_{FM}	conv4	78.05	94.45
L_{SR}	conv4	79.10	94.99
$L_{FM}+L_{SR}$	conv4	**79.58**	**95.21**
$L_{FM}+L_{SR}$	conv2	77.03	93.94
$L_{FM}+L_{SR}$	conv3	77.34	94.22
$L_{FM}+L_{SR}$	conv2+3+4	79.43	94.80

Table 4.2 Effect of proposed losses (L_{FM} and L_{SR}) and position of distillation on the test set of CIFAR100.

Where should the losses be applied? The proposed losses can be applied at other layers of the networks too. This is straightforward for L_{FM}. We can also extend L_{SR} to more layers, by transferring the mean feature of the student at each layer to the corresponding layer of the teacher using an AdaIN layer [60]. On one hand, applying the losses early in the network could ensure that the subsequent layers receive "better" features. On the other hand, features produced by early layers are not specialised to a particular class. Thus, applying the distillation losses towards the end of the network, where the activations encode discriminative, task-related features should lead to potentially stronger models. The results from Table 4.2 (last 3 rows) confirm our hypothesis: By applying the loss at multiple points in the network actually rather hurts accuracy.

Different losses for L_{SR}: Here, we evaluate different losses for L_{SR}. The following loss functions are compared:

1. L2 loss: $L_{SR-L2}(p,q) = \|p-q\|^2$.
2. Cross Entropy loss (CE) with label y: $L_{SR-CE}(q,y) = \mathscr{H}(q,y)$.

Method	KL div. with teacher	Cross-entropy with label	Top-1 (%)
Student	0.5964	0.9383	76.97
KD	0.5818	0.9492	78.35
AT	0.5406	0.9049	78.06
L_{FM}	0.5701	0.8980	78.05
L_{SR}	0.4828	0.8418	79.10
L_{FM}+L_{SR}	**0.4597**	**0.8247**	**79.58**

Table 4.3 KL divergence between teacher and student, and cross-entropy between student and ground truth on the test set of CIFAR100. Teacher's top-1 accuracy is 79.50%.

3. KL loss with temperature τ [55]: $L_{SR-KL}(p,q) = KL(q/\tau, p/\tau)$.

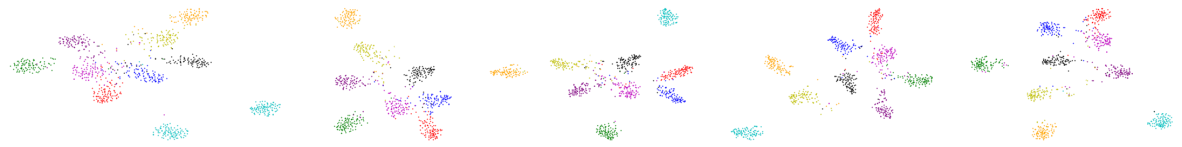
The results, presented in Table 4.4, show that all loss functions offer significant improvement gains while L_{FM}+L_{SR-L2} achieves the best accuracy. Therefore, in our paper, L_{SR-L2} is used in all cases.

Method	Top-1(%)	Top-5(%)
Student: WRN-16-4	76.97	93.89
Teacher: WRN-40-4	79.50	94.57
L_{FM}+L_{SR-L2}	**79.58**	**95.21**
L_{FM}+L_{SR-CE}	78.80	95.13
L_{FM}+L_{SR-KL}	79.04	95.12

Table 4.4 Evaluation of different loss functions for L_{SR} in terms of Top-1 accuracy on CIFAR100.

Teacher-student similarity: The overall aim of knowledge distillation is to make the student mimic the teacher's output, so that the student is able to obtain similar performance to that of the teacher. Therefore, to see how well the student mimics the teacher, we measured the similarity between the teacher's and student's outputs using (a) the KL divergence between the teacher's and student's outputs, and (b) the cross-entropy loss between the student's predictions and the ground truth labels.

From Table 4.3, it can be observed that KD [55] reduces the KL divergence with the teacher's output offering $\sim 1.5\%$ accuracy gain. AT [204] also decreases the KL divergence with the teacher's output offering a smaller accuracy gain of $\sim 1.0\%$. Moreover, both proposed losses L_{FM} and L_{SR} and their combination L_{FM}+L_{SR} show considerably high similarity compared to the KD and AT. This similarity is one of the main reasons for the improved student's accuracy offered by our method.



(a) Student (b) L_{FM} (c) L_{SR} (d) $L_{FM}+L_{SR}$ (e) Teacher

Fig. 4.2 Visualization of h_S and h_T on the test set of CIFAR100. Better viewed in color.

Method	Student	L_{FM}	L_{SR}	$L_{FM}+L_{SR}$
L2 Distance	1.48	1.33	1.07	**1.01**
NMI (%).	77.20	78.31	79.35	**79.85**
Top-1(%).	76.97	78.05	79.10	**79.58**

Table 4.5 L_2 Distance $\|h_T - h_s\|^2$, and NMI calculated on the test set of CIFAR100.

Representations distance: Table 4.5 shows the calculated L_2 distance between the teacher and student representations h_T and h_S. The results, presented in Table 4.5, clearly show that both L_{FM} and L_{SR} narrow the distance, with their combination being the closest to the teacher.

Normalized Mutual Information (NMI): Moreover, we calculated the NMI [98] which is a balanced metric that can be used to determine the quality of feature clustering. The results, presented in Table 4.5, show that $L_{FM}+L_{SR}$ has the highest NMI score meaning that the features are better clustered. Qualitative results are shown in Figure 4.2 which visualizes the features h_S and h_T. It can be observed that $L_{FM}+L_{SR}$ is able to learn more discriminative features, which also correlates with quantitative accuracy gains.

Transferability of representations: Following [153], this section aims to compare the representational power of the learned student's representation h_s. To this end, we trained the student on CIFAR100, and then used it as a frozen feature extractor on top of which we train a linear classifier for two datasets: STL10 [27] and CIFAR100. We compare the transferability of KD, CRD, L_{FM}, L_{SR}, and $L_{FM}+L_{SR}$. The superiority of the proposed losses over KD on STL is evident. Importantly, L_{SR} largely outperforms KD which confirms our analysis of Eqs. (4.5) and (4.6). The best results on STL are obtained by CRD. However, on CIFAR100, which is the target distillation dataset our method outperforms CRD.

Student	Dataset	KD	CRD	L_{FM}	L_{SR}	$L_{FM}+L_{SR}$
WRN16-4	STL10	68.75	**72.45**	69.3	71.44	72.17
WRN16-4	CIFAR100	78.28	78.46	77.95	79.03	**79.34**
MobileNetV2	STL10	62.17	**69.74**	66.12	68.23	68.95
MobileNetV2	CIFAR100	69.17	70.68	70.66	71.00	**71.63**

Table 4.6 Transferability of representations from CIFAR100 to STL10 and CIFAR100 by freezing f^S and training a linear classifier on top. Top-1 (%) accuracy is provided.

4.4.2 Comparison with State-of-the-Art

We thoroughly evaluated the effectiveness of our method across multiple (a) network architectures (ResNet [52], Wide ResNet [203], MobileNetV2 [133], MobileNet [57]) with different teacher-student capacities, (b) datasets (CIFAR10/100, ImageNet, MegaFace), and (c) domains (real-valued and binary networks). We denote with ResNetN a Residual Network with N convolutional layers [52]. We denote with WRN-*D*-*k* a WRN architecture with D layers and an expansion rate of k [204].

For the above mentioned settings, we compare our method with KD [55] and AT [204], and the more recent methods of OFD [53], RKD [115], CRD [153]. For traditional KD [55], we set $\alpha = 0.9$ and $T = 4$. For AT [204], as in [204, 156], we set the weight of distillation loss to 1000. We note, that for our experiments, the AT loss is added after each layer group for WRN and the last two groups for ResNet as in [204]. Following OFD [53], we set the weight of distillation loss to 10^{-3}. For RKD [115], we set $\beta_1 = 25$ for distance, and $\beta_2 = 50$ for angle, as described in [115, 153].

Overview of results: From our experiments, we conclude that our approach offers consistent gains across all of the above scenarios, outperforming all methods considered for all settings. Notably, our method is particularly effective for the most difficult datasets (i.e. CIFAR100, ImageNet-1K and MegaFace).

Image Classification on CIFAR10 Dataset

Implementation details: CIFAR10 is a popular image classification dataset consisting of 50,000 training and 10,000 testing images equally distributed across 10 classes. All images are of resolution 32×32px. Following [204], during training, we randomly cropped and horizontally flipped the images. The ResNet models were trained for 350 epochs using SGD. The initial learning rate was set to 0.1, and then it was reduced by a factor of 10 at epochs 150, 250 and 320. Similarly, the WRN models were trained for 200 epochs with a learning rate of 0.1 that was subsequently reduced by 5 at epochs 60, 120 and 160. In all experiments, we set the dropout rate to 0. We did not compare with CRD [153] on CIFAR10 because, in our experiments, we found that their parameter setting (used in their paper for CIFAR100 and ImageNet-1K) does not obtain good performance on CIFAR10.

Results: Top-1 performance of our method on CIFAR10 is shown in Table 4.7. We tested three cases representing different network architectures for student and teacher networks: the first two experiments are with WRNs. The following three experiments are with ResNets. In the last experiment, teacher and student have different network architectures. Specifically, our method improves student network's top-1 performance from 91.04% to 92.95% on pair WRN-16-1 as student, WRN-16-2 as teacher, from 93.98% to 94.66% on pair WRN-16-2 as student, WRN-40-2 as teacher, from 87.78% to 89.02% on pair ResNet8 as student, ResNet26 as teacher, from 91.59% to 92.70% on pair ResNet14 as student, ResNet26 as teacher, from 93.35% to 93.92% on pair ResNet-18 as student, ResNet34 as teacher, from 91.04% to 92.94% on pair WRN-16-1

Student(Params)	Teacher(Params)	Student	KD [55]	AT [204]	OFD [53]	RKD [115]	Ours	Teacher
WRN-16-1 (0.18M)	WRN-16-2 (0.69M)	91.04	92.57	92.15	92.28	92.51	**92.95**	93.98
WRN-16-2 (0.69M)	WRN-40-2 (2.2M)	93.98	94.46	94.39	94.30	94.41	**94.66**	95.07
ResNet8 (0.08M)	ResNet26 (0.37M)	87.78	88.75	88.15	87.49	88.50	**89.02**	93.58
ResNet14 (0.17M)	ResNet26 (0.37M)	91.59	92.57	92.11	92.51	92.36	**92.70**	93.58
ResNet18 (0.7M)	ResNet34 (1.4M)	93.35	93.74	93.52	93.80	92.95	**93.92**	94.11
WRN-16-1 (0.18M)	ResNet26 (0.37M)	91.04	92.42	91.32	92.47	92.08	**92.94**	93.58

Table 4.7 Top-1 accuracy (%) of various knowledge distillation methods on CIFAR10.

as student, ResNet26 as teacher. Overall, our method achieves the best results for all cases, with KD [55] closely following.

Image Classification on CIFAR100 Dataset

Implementation details: CIFAR100 [76] is a popular image classification dataset consisting of 50,000 training and 10,000 testing images equally distributed across 100 classes. Here, we used a standard data augmentation scheme [204] including padding 4 pixels prior to random cropping and horizontal flipping. We used SGD with weight decay 5e-4 and momentum 0.9. The batch size was set to 128. The learning rate was set to 0.1; then decayed by 0.1 at epochs 100, 150, until training reached 200 epochs [53]. We experimented with several student-teacher network pairs using different structures. Experiments are grouped in three sets. The first shows performance for different teacher and student capacities using WRNs: poor student - good teacher (WRN-16-2; WRN-40-4), descent student - good teacher (WRN-10-10; WRN-16-10); good student - good teacher (WRN-16-4; WRN-40-4). In the second set, we show that these results hold when using a different architecture, ResNet in this case. The final set is designed to show performance when teacher and student have different architectures (MobileNetV2, ResNet and WRN).
Results: Top-1 performance of our method is shown in Table 4.8. We observe that for almost all configurations, our method achieves consistent and significant accuracy gains over prior work. Furthermore, it is hard to tell which is the second best method as the remaining methods have their own advantages for different configurations. For WRN experiments, OFD ranks second. For ResNet and mixed structure experiments, CRD ranks second. We also provide the results by combining our method with KD and AT. Based on the order in 4.8 KD+Ours:74.97%, 79.00%, 78.84%, **70.41%**, 73.46%, 77.64%, 74.90%, 71.08%, 70.85%. AT+Ours: 75.01%, 79.09%, 77.79%, 69.41%, 73.17%, 77.48%, 74.71%, 70.70%, 70.63%. Further improvements could be obtained by combining our method with others but this requires a comprehensive investigation which goes beyond the scope of this thesis.

Image Classification on ImageNet-1K Dataset

Implementation details: ImageNet-1K [130] considered one of the most heavily benchmarked datasets in computer vision consists of 1.2M training images and 50K validation images with

Student (Params)	Teacher (Params)	Student	KD [55]	AT [204]	OFD [53]	RKD [115]	CRD [153]	Ours	Teacher
WRN-16-2 (0.70M)	WRN-40-4 (8.97M)	72.70	74.52	74.33	75.57	74.23	75.27	**75.96**	79.50
WRN-16-4 (2.77M)	WRN-40-4 (8.97M)	76.97	78.35	78.06	79.29	78.38	78.83	**79.58**	79.50
WRN-10-10 (7.49M)	WRN-16-10 (17.2M)	76.27	78.20	76.44	78.72	77.84	78.35	**79.17**	79.77
ResNet10 (0.34M)	ResNet34 (1.39M)	68.42	69.18	68.49	68.94	68.70	**70.24**	69.91	72.05
ResNet18 (0.75M)	ResNet50 (1.99M)	71.07	73.41	71.90	72.79	70.93	73.23	**73.47**	73.31
ResNet10 (4.95M)	ResNet34 (21.33M)	75.01	77.35	76.87	77.35	77.46	77.37	**77.90**	78.44
WRN-16-2 (0.70M)	ResNet34 (21.33M)	72.70	73.95	72.32	74.78	73.91	74.88	**75.38**	78.44
MobileNetV2 (2.37M)	ResNet34 (21.33M)	68.42	69.36	68.60	69.45	68.75	71.36	**71.58**	78.44
MobileNetV2 (2.37M)	WRN-40-4 (8.97M)	68.42	69.15	68.95	70.08	68.19	71.46	**71.82**	79.50

Table 4.8 Top-1 accuracy (%) of various knowledge distillation methods on CIFAR100.

Student (Params)	Teacher (Params)		Student	KD [55]	AT [204]	OFD [53]	RKD [115]	CRD [153]	Ours	Teacher
ResNet18 (11.69M)	ResNet34 (21.80M)	Top-1	70.04	70.68	70.59	71.08	71.34	71.17	**71.73**	73.31
		Top-5	89.48	90.16	89.73	90.07	90.37	90.13	**90.60**	91.42
MobileNet (4.23M))	ResNet50 (25.56M)	Top-1	70.13	70.68	70.72	71.25	71.32	71.40	**72.49**	76.16
		Top-5	89.49	90.30	90.03	90.34	90.62	90.42	**90.92**	92.86

Table 4.9 Comparison with state-of-the-art on ImageNet.

1K classes. Images are cropped to 224×224 pixels for both training and evaluation. We used SGD with Nesterov momentum 0.9, weight decay $1e-4$, initial learning rate 0.2 which was then dropped by a factor of 10 every 30 epochs, training in total for 100 epochs (for CRD we trained with 10 more epochs as suggested by the authors). The batch size was set to 512. For simplicity and to enable a fair comparison, we used pretrained PyTorch models [119] as teacher networks [53, 153].

Our experiments include two pairs of networks which are popular settings for ImageNet [130]. The first is distillation from ResNet34 to ResNet18 and the second one is distillation from ResNet50 to MobileNet [57]. Note that, following [153] on ImageNet, for KD, we set the weight for KL loss to 0.9, the weight for cross-entropy loss to 0.5 which helps to obtain better accuracy. **Results:** Our results are presented in Table 4.9. Again, we observe that our method achieves significant improvements over all methods. To be specific, our method improves the ResNet18 network performance from 70.04% to 71.73% on top-1 accuracy, from 89.48% to 90.60% on top-5 accuracy with ResNet34 as teacher. Also, it improves MobileNet performance from 70.13% to 72.49% on top-1 accuracy, from 89.49% to 90.92% on top-5 accuracy with ResNet50 as teacher. Moreover, there is no method which is consistently second: for ResNet34 to ResNet18 experiment, RKD is the second best, with top-1 71.34% and top-5 90.13%, while for ResNet50 to MobileNet, CRD is the second best, with top-1 71.40% and top-5 90.42%. Notably, for the latter experiment, CRD reduces the gap between the teacher and the student by 1.27%, while our method narrows it by 2.36%. Overall, our results on ImageNet validate the scalability of our method, and show that, when applied to a large-scale dataset, we achieve an even more favourable performance compared against competing methods.

Image Classification with Binary Networks Distillation

Implementation details: Training highly accurate binary neural networks (i.e. the most extreme case of quantization) is a very challenging task [125, 11], and to this end, knowledge distillation appears to be a promising direction. In this section, we present results by applying distillation for the task of training binary student networks guided by real-valued teacher networks. The network architecture is kept the same for both the student and the teacher in this case: specifically, we used a ResNet using the modifications described in [11]. For training, we used Adam as the optimizer with initial learning 0.001 which is reduced by a factor of 10 at epochs 150, 250, and 320, training in total for 350 epochs on CIFAR100 and with initial learning 0.002 which is reduced by a factor at epochs 30, 60, and 90, training in total for 100 epochs on ImageNet-1K.
Results: Table 4.10 presents our results. Again, we observe that our method outperforms all methods considerably, showing that it can effectively transfer knowledge between different domains. Note that, it was not clear to us where to place the distillation position for OFD, so although we included our result for this method, we emphasize that this result might be suboptimal.

Dataset	Method	Binary	KD [55]	AT [204]	OFD [53]	RKD [115]	CRD [153]	Ours	Real
CIFAR100	ResNet34	65.34	68.65	68.54	66.84	68.61	68.78	**70.50**	75.08
ImageNet-1K	ResNet18	56.70	57.39	58.45	55.74	58.84	58.25	**59.57**	70.20

Table 4.10 Real-to-binary distillation results on CIFAR100: a real-valued teacher ResNet34 is used to distill a binary student. Real-to-binary distillation results on ImageNet-1K: a real-valued ResNet18 is used to distill a binary student. OFD result might be sub-optimal.

Face Recognition on MegaFace Dataset

Implementation details: We conducted the large scale MS1MV2 experiment reported in [34]. For training, we used MS1MV2, a refined (cleaned) version of MS-Celeb-1M [47, 177]. For testing, we used the refined MegaFace [34, 47, 72] for evaluating the performance of face recognition at million-level distractors. The gallery set of MegaFace consists of 1M images from 690K subjects, and the probe set consists of 100K photos from 530 unique individuals from FaceScrub. We used ResNet101 as the teacher network, and MobileFaceNet as the student network. The network settings are the same as in [34, 23]. The MobileFaceNet structure is in Table 4.11. For training, we set the initial learning rate to 0.1, and multiplied it by 0.1 at 100K, and 160K iterations, training in total for 180K iterations. We used SGD as optimizer and set momentum to 0.9, and weight decay to $5e-4$. The batch size was set to 512.
Results: Rank-1 identification rate at different number of distractors is used as metric for evaluation. Our results are shown in Table 4.12 and Figure 4.3. It can be seen that, on this challenging face recognition task, our method improves about 0.7% in 1:1 verification and 1% at rank-1 identification rate with 1M distractors.

Input	Output	operator	in_channel	groups	out_channel	number	stride
112x112	56x56	conv_block	3	1	64	1	2
56x56	56x56	conv_block	64	64	64	1	1
56x56	28x28	bottleneck	64	128	64	1	2
28x28	28x28	bottleneck	64	128	64	4	1
28x28	14x14	bottleneck	64	256	128	1	2
14x14	14x14	bottleneck	128	256	128	6	1
14x14	7x7	bottleneck	128	512	128	1	2
7x7	7x7	bottleneck	128	256	128	2	1
7x7	7x7	conv_block	128	1	512	1	1
7x7	1x1	conv_block	512	512	512	1	1
1x1	512	Linear	512	1	512	1	1

Table 4.11 MobileFaceNet architecture [23]. Each line describes a sequence of operations. Convolutions in the conv_blocks use 3×3 kernels except the last conv_block using 7×7. Each line denotes input feature's spatial size (width $\times$height), output feature map's spatial size, input feature map's channel, output feature map's channel, repeated number of this operation and stride size in convolutional layer. Groups represents the extension channel number and also the group number in 3×3 convolution in bottleneck.

Student (Params)	Teacher (Params)		Student	KD [55]	AT [204]	RKD [115]	PKT [117]	Ours	Teacher
MobileFaceNet (1.2M)	ResNet101 (65.12M)	Ver %	93.44	92.86	93.55	93.37	93.25	**94.17**	98.56
		Id %	92.28	90.91	92.46	92.34	92.38	**93.26**	98.82

Table 4.12 Face identification and verification evaluation of different methods on MegaFace Challenge1 using FaceScrub as the probe set. "Ver" refers to the face verification TAR (True Acceptance Rate) at 10^{-6} FAR (False Acceptance Rate) and "Id" refers to the rank-1 face identification accuracy with 1M distractors.

More comparisons on LFW, AgeDB, CPLFW and CALFW are shown in Table 4.13. Our proposed method consistently outperforms other distillation methods on these face verification benchmarks.

Facial Landmark Detection on WFLW

Given a face image, the task is to localise a set of facial landmarks in terms of their (x,y) coordinates. This is often solved by using a CNN to directly regress the (x,y) coordinates of the facial landmarks. In order to show the suitability of our method for this problem, we use the WFLW [176] dataset, which is one of the hardest benchmarks for this task. Performance is measured in terms of the Normalised Mean Error (lower is better), which is the standard metric for the problem. In our experiment, we use a ResNet10 as the teacher and a ResNet10 as the student. The results, shown in Table 4.14, confirm the superior performance of our method when compared to other state-of-the-art methods.

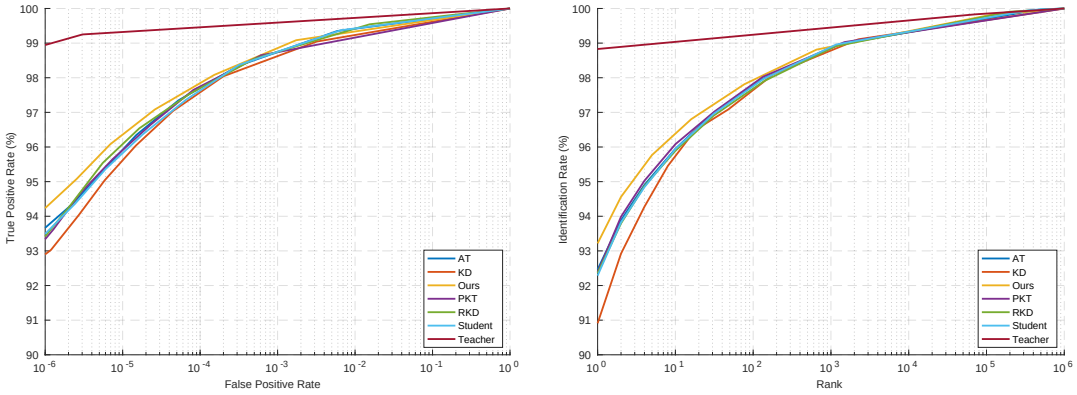


Fig. 4.3 Verification ROC and identification CMC curves of all distillation methods. Results are evaluated on refined MegaFace dataset.

Method	Verification Evaluation (%)			
	LFW	AgeDB	CPLFW	CALFW
Student	99.27	96.21	89.75	95.21
Teacher	99.83	98.40	93.30	96.11
KD [55]	**99.61**	96.21	90.19	95.11
AT [204]	99.60	96.45	89.38	95.16
RKD [115]	99.41	96.06	89.43	95.28
PKT [117]	99.55	96.33	89.93	95.21
Ours	**99.61**	**96.78**	**90.20**	**95.46**

Table 4.13 Verification results on LFW, AgeDB, CPLFW, CALFW

Student(Params)	Teacher(Params)	-	Student	KD [55]	RKD [115]	PKT [117]	L_{FM}	AT [204]	Ours	Teacher
ResNet10(7.25M)	ResNet50(26.25M)	NME	7.43	7.32	6.94	7.09	7.14	6.96	**6.81**	6.38

Table 4.14 Facial landmark detection with ResNet50 as teacher and ResNet10 as student. KD is adapted by using an L2 loss instead of a KL loss to measure the discrepancy between the teach and student predictions. We use ResNet10 as student because ResNet18 performance is close to ResNet50.

4.5 Experiments on Unlabeled Datasets

4.5.1 Image Classification on CIFAR100 Dataset

In this section, we investigate how to use unlabeled data from TinyImageNet [1] to boost current student's performance on CIFAR100 [76] under current knowledge distillation frameworks. In the remaining part of this section, we will first introduce the experiment setting, then investigate various factors affecting the overall performance, and finally provide more comparisons under different teacher-student pairs.

[1] https://tiny-imagenet.herokuapp.com/

Implementations: The CIFAR100 dataset, as the labeled dataset, is one of the most commonly used dataset for evaluating knowledge distillation methods. It consists of 100 semantic classes with a resolution of 32x32. In total, it has 50K images for the training set (named as *CIFAR100-Train*) with 500 images per class and 10K for the test set (named as *CIFAR100-Test*) with 100 images per class. TinyImageNet, serving as the unlabeled dataset here, is a subset of ImageNet-1K [130], consisting of 200 classes and each class has 500 training images, 50 validation images, and 50 test images with size 64x64 pixels. In our case, we use its 100K training set as the unlabeled dataset, and name it as *Tiny-Train* for simplicity.

Generally, our experiments build on open source code from [153][2] to allow a direct comparison with several works from the literature. For the training setting, we use synchronous stochastic gradient descent as an optimizer with momentum 0.9, weight decay 5e-4. Except ShuffleNetV1/V2 and MobileNetV2 using 0.01 as starting learning rate, other models usually train from 0.05, then decays by 0.1 at epochs 150, 180, 210 until 240 epochs.

Teacher	ResNet32x4	79.42	WRN40-2	75.61	ResNet32x4	79.42
Student	ResNet8x4	72.50	WRN40-1	71.98	ShuffleNetV1	70.50
	D	D+U	D	D+U	D	D+U
KD [55]	73.33	74.68↑	73.54	75.08↑	74.07	76.52↑
FitNet [129]	73.44	73.30↓	72.24	71.43↓	73.59	72.83↓
AT [204]	73.44	71.75↓	72.77	73.11↑	71.73	72.82↑
SP [156]	72.94	72.10↓	72.43	73.02↑	73.48	76.01↑
CC [120]	72.97	71.96↓	72.21	70.64↓	71.14	70.85↓
VID [1]	73.09	73.48↑	72.30	73.06↑	73.38	75.80↑
RKD [115]	71.90	72.50↑	72.22	72.99↑	72.28	73.38↑
PKT [117]	73.64	75.24↑	73.45	74.29↑	74.10	76.50↑
AB [54]	73.17	72.34↓	72.38	72.88↑	73.55	73.33↓
FT [73]	72.86	71.57↓	71.59	71.47↓	71.75	72.81↑
NSP [62]	73.30	72.06↓	72.24	72.11↓	74.12	-
CRD [153]	75.51	75.99↑	74.14	75.15↑	75.11	77.00↑
Ours	75.92	76.24↑	74.75	75.32↑	76.40	77.70↑
	D+U+CS	D+U+MP	D+U+CS	D+U+MP	D+U+CS	D+U+MP
KD	72.82↓	76.07↑	74.34↓	75.34↑	75.64↓	77.45↑
CRD	74.71↓	76.80↑	73.62↓	75.48↑	75.79↓	77.72↑
Ours	75.66↓	**77.07**↑	73.70↓	**75.66**↑	76.34↓	**78.32**↑

Table 4.15 Classification performance (%) of student models on CIFAR100. D means training on *CIFAR100-Train*. D+U means training on *CIFAR100-Train* and *Tiny-Train*. CS denotes consistency loss and MP denotes training with 4.2.2. Average over 5 independent runs.

[2] https://github.com/HobbitLong/RepDistiller

Comparison under different knowledge distillation frameworks: We totally train 13 state-of-the-art methods (*eg.* KD [55], FitNet [129], AT [204], SP [156], CC [120], VID [1], RKD [115], PKT [117], AB [54], FT [73], NSP [62], CRD [153], Ours) following Eq. 4.9 with different teacher and student pairs (ResNet32x4 as teacher, ResNet8x4 as student; WRN40-2 as teacher, WRN40-1 as student; ResNet32x4 as teacher, ShuffleNetV1 as student).

The results are shown in Table 4.15. Specifically, `D` represents results from distillation loss on labeled data and `D+U` represents methods distillation loss from both on labeled dataset and unlabeled dataset). We can see that KD, VID, RKD, PKD, CRD and Ours consistently gain performance from extra *Tiny-Train* while FitNet, CC degrade in all settings. In addition, AT, SP, AB and FT vary among different student-teacher pairs. Overall, KD, CRD and Ours are still superior to others among all settings which is consistent with their performance trained on labeled dataset alone. It is worth mentioned that KD is the biggest beneficiary from unlabeled data with top-1 accuracy increasing by $\sim 1.3\%$, $\sim 1.5\%$, $\sim 2.5\%$. To some extent, it explains the effectiveness of pseudo labelling in semi-supervised learning with previous predictions as labels to train the subsequent model. However, for the final performance comparison, CRD and Ours still preserve their superiority to KD. In conclusion, the above experiments suggest that the improvement with unlabeled datasets depends on the relative performance of student and teacher models trained on the original supervised task (i.e., training only on the labeled datasets) and knowledge formation defined in distillation.

Amount of unlabeled datasets: So far, we use the entire *Tiny-Train* in student learning. To investigate the effect from the dataset size, we perform experiments with varying amount of *Tiny-Train* on two choices: one way is to randomly select 25%, 50%, 75% of *Tiny-Train*, and the other way is to select 25%, 50%, 75% of *Tiny-Train* based on highest confidence provided teacher network. For this purpose, we use softmax to normalize 100-dim prediction from the teacher network. Then its largest response indicating the prediction confidence is used to do sample selection.

The result is shown in Figure 4.4. Overall, we find that the more unlabeled data used, the more gains the student network will achieve. Also, small percentage of unlabeled data will also benefit the student's performance, and randomly selected images at low portion are better than score-based selection. We argue this is caused by class bias from the score-based selection. We agree that here using all images from unlabeled data is sub-optimal. We emphasize that these results only apply to our specific labeled (*CIFAR100-Train*) and unlabeled dataset (*Tiny-Train*) selection and may not provide general insight that any unlabeled dataset will improve student learning. Also we know by manual inspection that images from *Tiny-Train* are closely related to CIFAR100 and the improvement shows that *Tiny-Train* is high-quality compensation for *CIFAR100-Train*. For instance, they share same categories like "n02481823":21 (chimpanzee), "n02129165":43 (lion), "n04465501":89 (tractor), "n07747607":53 (orange), "n03649909":41 (lawn_mower0), "n02206856":6 (bee). However, as the class definition for CIFAR100 and TinyImagenet are different, it is hard to find all same classes.

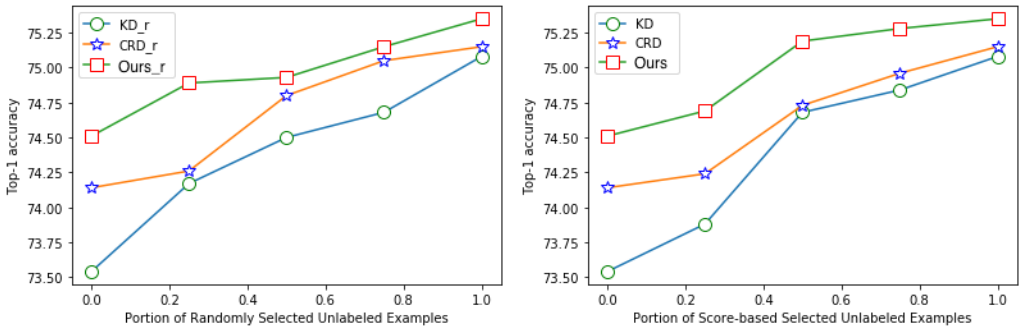


Fig. 4.4 Top-1 accuracy of KD, CRD, and Ours on CIFAR100 with 25%, 50%, 75%, and 100% of *Tiny-Train*. The left hand result is from randomly selection and the right hand result is from confidence-based selection.

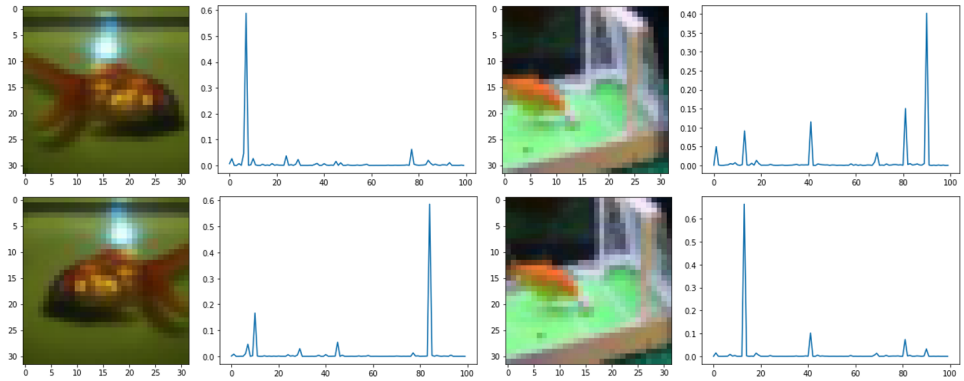


Fig. 4.5 Teacher network is not robust on the unlabeled data after augmentation. Each column represents a couple of augmented inputs. For left column, teacher's predictions are"table" and "beetle" respectively. For right column, teacher's predictions are "train" and "bus", respectively.

Effect of consistency loss: Here, We try to explore consistency loss on usage of the unlabeled data. In practice, the stochastic data augmentation made up of randomly 4 pixels shifting and randomly horizon flipping is applied to $x \in \mathtt{U}$ to produce two augmented versions x_{u1} and x_{u2}. x_{u1} is fed to teacher network to get prediction y^T_{u1}. Predictions from student network for x_{u1} and x_{u2} are y^S_{u1} and y^S_{u2} respectively. The consistency loss is added to enforce similarity on y^S_{u1} and y^S_{u2}.

As a result, we find that consistency loss degrades the model performance (see in Table 4.15) in all cases. Similar observations can be found in [81, 63]. These works suffer the same problem since they depend on a fixed pre-trained model instead of a converged model with high accuracy to generate pseudo labels. In essence, in the unlabeled dataset, the teacher network is not trained to satisfy the data augmentation consistency constraint on unseen label data. The more intuitive analysis is illustrated in Figure 4.5. For two augmentation versions x_{u1} and x_{u2}, the teacher's prediction y^T_{u1} and y^T_{u2} are not consistent. After doing a statistic analysis, we have found that there are roughly 64% images with inconsistent predictions from teacher.

Effect of MixUp: We explore the effect of mixup in Section 4.2.2 for knowledge distillation. As is shown in Table 4.15, this trick further improves the student network's performance in all settings. KD, CRD and ours roughly gain 2% after doing mixup.

Teacher	vgg13	74.64	ResNet50	79.34	WRN40-2	75.61	ResNet32x4	79.42
Student	vgg8	70.36	MobileNetV2	64.60	ShuffleV1	70.50	ShuffleNetV2	71.82
	D	D+U+MP	D	D+U+MP	D	D+U+MP	D	D+U+MP
KD [55]	72.98	74.71	67.35	70.55	74.83	77.40	74.45	78.05
CRD [153]	73.94	74.93	69.11	71.44	76.05	77.78	75.65	78.25
Ours	74.40	**75.03**	69.45	**71.46**	76.56	77.77	76.40	**78.53**

Table 4.16 Classification performance (%) of student models on *CIFAR-100-test*. Baseline is trained on `D`. `D+U+MP` is trained on whole dataset with individual knowledge distillation loss and mix up in Section 4.2.2. Average over 5 independent runs.

More comparisons are in Table 4.16. In all teacher-student pairs (vgg13 as teacher vgg8 as student, ResNet50 as teacher MobileNetV2 as student, ResNet32x4 as teacher ShuffleNetV2 as student), it is shown that by introducing unlabeled data in knowledge distillation, the gap between student and teacher is further narrowed. Besides, in baseline setting in which ResNet34x4 as teacher and ShuffleNetV2 as student, Ours is nearly 1% better than CRD. After introducing unlabeled data in training, both of them are improved and in addition, CRD is closer to Ours. It verifies that self-supervised training together with supervised contrastive learning is good at increasing network's feature representation ability.

4.5.2 Image Classification on ImageNet-1K-Sub Dataset

In this section, we investigate how to use unlabeled data from Yahoo Flickr Creative Commons 100 Million (YFCC-100M) [152] [152] to boost the student' performance on *ImageNet-1K-Sub* which is a subset of ImageNet-1K [130].

Implementations: ImageNet-1K [130] considered as one of the most heavily benchmark datasets in computer vision consists of 1.2M training images and 50K validation images with 1K classes. Images are cropped to 224x224 pixels for both training and evaluation. YFCC-100M is a public dataset of about 100 million images from Flickr website. It provides a rich resource over a large amount of visual concepts, and reflects well the user preferences on the Flickr platform. On the contrary to ImageNet-1K, YFCC-100M misses clean annotations. However, due to the large scale visual entities, it is popular to use it as extra compensation for ImageNet-1K [188, 121]).

We study factors like data filtering, balancing, ratios and training procedure in this section to find the best setting. In practice, we create a subset of ImageNet-1K named *ImageNet-500-500-train* by randomly selecting 500 classes and 500 images per class. The evaluation is done on *ImageNet-500-50-eval* with same 500 classes and 50 images per class from original evaluation dataset. The models are trained with SGD as optimizer, weight decay $1e-3$ and a standard learning rate of 0.1, and decayed by 0.1 every 30 epochs for total 100 epochs. The batch size is set to 256.

Results We conduct ablation studies on the teacher-student pair: ResNet50 as teacher, ResNet18 as student. Teacher network trained on *ImageNet-500-500-train* achieves Top1 accuracy 69.10% on *ImageNet-529-50-eval*. The student network performance trained with our proposed method

achieves Top1 accuracy 66.47% on *ImageNet-529-50-eval*. The main framework is inherited from Section 4.5.1. It is composed of labeled loss (cross-entropy loss with labels, knowledge distillation loss with teacher's prediction), unlabeled loss (knowledge distillation loss with teacher's prediction) and mixup loss (KL divergence loss on mixed labeled and unlabeled dataset).

We consider the following cases for unlabeled data from YFCC-100M:

- AS1: Randomly select 500x500 images from YFCC-100M dataset as unlabeled dataset. The rate between labeled data and unlabeled data is 1:1. In this case, unlabeled dataset inevitably contains a large fraction of images not belonging to 500 classes. Finally, the student gains 0.55% in total which indicates randomly selected samples is still beneficial in model performance.

- AS2: Filter the YFCC-100M dataset with teacher's confidence 0.3 and randomly select 500x500 images (without considering the classes) from the filtered YFCC-100M. The rate between labeled data and unlabeled data maintains 1:1. The model gains another 0.65% in total which indicates that score-based selection in this case is superior to random selection. This is not contrastive to the finding in 4.5.1 as YFCC-100M is far from ImageNet in data distribution.

- AS3: Filter the YFCC-100M dataset with teacher confidence 0.3 and randomly select 500 images per class from the filtered YFCC-100M. The rate between labeled data and unlabeled data is 1:1. It gains another 0.58% which verifies the effectiveness of filtering and balancing in extra data selection.

- AS4: Randomly select 500 images per class from the rest ImageNet-1K training set as unlabeled data. The rate between labeled data and unlabeled data maitains 1:1. In this case, unlabeled data shares the same distribution as labeled data. The performance is 68.78% and it is treated as the upper bound of variants AS1-AS4.

- AS5: Filter the YFCC-100M dataset with teacher confidence 0.3 and randomly select 1500 images per class from the filtered YFCC-100M. The rate between labeled data and unlabeled data is 1:3. The final performance is close to AS3. We assume it is caused by incomplete training in the current defined 100 epochs.

- AS6: Use the unlabeled from AS5 but increase the whole training epochs to 300. It increases by over 1% compared to AS5. It shows that more training time is necessary when large unlabeled data are used.

- AS7: Use full training set for 500 classes to train a supervised student model. Inevitably, Using unlabeled data will be sub-optimal, compared to using full clean dataset with annotations. The performance of the supervised training can be seen as an upper bound of the learning with unlabeled data.

Methods	D:U	Epochs	Filtered	Balanced	Accuracy
AS1	1:1	100	No	No	67.02
AS2	1:1	100	Yes	No	67.67
AS3	1:1	100	Yes	Yes	68.25
AS4	1:1*	100	-	Yes	68.78
AS5	1:3	100	Yes	Yes	68.34
AS6	1:3	300	Yes	Yes	69.51
AS7	x	100	No	No	**71.42**

Table 4.17 Ablation studies on various design choices: labeled and unlabeled data rate, training epochs, data filtering, class balancing. "*" denotes that unlabeled data come from rest of training set.

The comparison is in Table 4.17.

Therefore, we use the best setting found in Table 4.17 to do more comparison with state-of-the-art methods in Table 4.18. Specifically, we use ResNet50 as teacher to guide two students with different structures: ResNet18 and MobileNet.

D							D+U+MP		
Teacher	Student	Teacher	Student	KD [55]	CRD [153]	Ours	KD [55]	CRD [153]	Ours
ResNet50	ResNet18	69.10	63.49	64.4	65.67	66.47	68.23	69.00	**69.51**
ResNet50	MobileNet	69.10	65.10	65.73	66.93	67.96	70.25	71.05	**71.32**

Table 4.18 Top-1 accuracy (%) of various knowledge distillation methods on *ImageNet-529*. ResNet50 as teacher and ResNet18 as student.

4.5.3 Face Recognition on UmdFace Dataset

Implementation details: In this section, we try to improve the face recognition models' performance trained on UmdFace [4] dataset with the extra dataset from VGGFace2[13]. UMDFace datset has 8.2K identities with in total 0.4M images. VGGFace2 [13] (an improved version of VGGFace [116]), containing 3.31M images of 9,131 subjects with large variations in pose, age, illumination, ethnicity and profession. VGGFace2 serves as unlabeled dataset in our setting. We followed standard practices in face recognition [170, 94, 162, 34] to crop a face image of 112×112 which was normalized to [-1, 1]. The training faces are randomly flipped for data augmentation.

We used ResNet101 as the teacher network, and ResNet18 as the student network. The network settings are the same as in [34, 23]. For training, we set the initial learning rate to 0.1, and multiplied it by 0.1 at epoch 16, 20, 24, 28, training in total for 32 epochs. We used SGD as optimizer and set momentum to 0.9, and weight decay to $5e-4$. The batch size is set to 512. We only evaluate our proposed softmax regression representation learning method here as above sections have verified its superiority over others.

Results: We report the verification accuracy on the LFW [59], CFP-FP[136], AgeDB [104], CPLFW [222], CALFW [223]. CPLFW and CALFW are variants of LFW. In Table 4.19, we can conclude that usage of unlabeled data consistently improves the student performance from a large margin.

Method	Verification Evaluation (%)				
	LFW	CFP-FP	AgeDB	CPLFW	CALFW
Student	98.88	89.69	92.63	85.46	93.05
Teacher	99.30	94.25	94.11	88.73	94.28
Ours (trained on D)	98.93	91.34	93.68	86.46	93.28
Ours (trained on D+U)	**99.28**	**92.91**	**93.91**	**88.25**	**93.56**

Table 4.19 ResNet101 as teacher and ResNet18 as student. Labeled data D is UMDFace and unlabeled data U is VGGFace2. Verification results are on LFW, CFP-FP, AgeDB, CPLFW, CALFW.

4.6 Conclusion

We presented a method for knowledge distillation that optimizes the output feature of the penultimate layer of the student network and hence is directly related to representation learning. A key to our method is the newly proposed Softmax Regression Loss which was found necessary for effective representation learning. We showed that our method consistently outperforms other state-of-the-art distillation methods for a wide range of experimental settings including multiple network architectures (ResNet, Wide ResNet, MobileNet) with different teacher-student capacities, datasets (CIFAR10/100, ImageNet, MegaFace), and domains (real-valued and binary networks).

The proposed method can be integrated to FAN-Face in chapter 3 to further improve small network's performance on unaligned face images. Also, it is a topic worth exploring that whether we can train student without FAN. That is to say, the teacher is trained with prior from FAN while the student is trained without it but with the shape correspondence knowledge in teacher's feature. We suppose this solution will work as the teacher can enable student to learn which feature to focus. Besides, the proposed knowledge distillation method can enhance the network performance after improving interpretability. In this case, we will obtain a explainable face recognition model with decent performance.

Chapter 5

Interpretable Face Recognition via Unsupervised Mixtures of Experts

This chapter is to learn visual attributes using dynamically-routed CNN with a supervised face recognition loss and an unsupervised clustering loss. The former loss is on the face embedding and the latter one is on the intermediate representation, specifically the routing vector in dynamic convolution. We observe that the routing vector in dynamic convolution is capable of partitioning data based on representational attributes but results in a noisy and scattered distribution. Based on these discoveries, we instantiate a group dynamic convolution to enhance the routing vector's discrimination capacity. Besides, we propose to leverage an unsupervised clustering loss to push the routing vectors to a fixed number of clusters in the feature space without damaging the recognition performance. In addition, due to the lack of labeled data, we annotate three datasets with 5 facial attributes (gender, age, race, pitch and yaw) and then explore routing vectors' behaviour on them with clustering analysis. We demonstrate that learned clusters are able to encode rich facial properties without requiring hand-designed attributes for supervision. Moreover, we present convincing results on the face inversion task that the routing vector provides auxiliary prior beneficial to producing images faithful to input.

5.1 Introduction

Facial attributes such as pose, age and demographics, provide valuable cues for understanding face images and in turn, also affect the performance of face analysis. Generally, the face recognition model improves performance when trained together with attribute prediction tasks [167, 124]. This observation confirms that there is a high correlation in both tasks in the shared space. With the help of auxiliary information provided by head pose estimation, [12] has bridged the discrepancy between the profile and frontal faces in the feature space and obtained a pose-robust recognition model. These beneficial insights have drawn researchers' attention and recent works are beginning to examine and explain these behaviours.

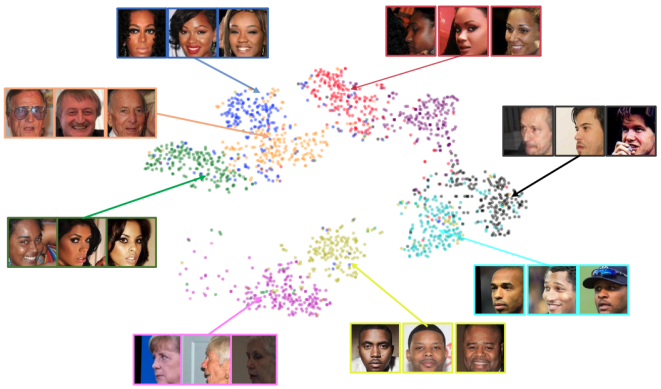


Fig. 5.1 2D feature space on routing vectors extracted from CNNs with dynamic convolution (DC:C=300 in Section 5.4 on Sub200 dataset in Section 5.4.1). Each color denotes a couple of faces with shared attributes. For example, the red color represents young black women with left profile face. It shows that routing vectors convey attribute information.

One line of research addresses the question of what is stored in the face embedding. These works [113, 114, 112, 151] first pre-define some characteristics like head position, social traits and train an estimation model on face representation to predict attributes and finally examine its performance to determine whether these attributes exist. As an alternative, [37] has leveraged another network to check features' expressive ability to inform us of the facial attributes. Another line of work is modifying or designing a new network structure where the internal representation works as a pattern detector, and then investigating the network's operation process [185, 74]. Chao *et al.* [185] have discovered that nodes in a tree-like network are sparsely activated by the sample's characteristics such as occlusion or viewpoints. Recently, after learning a final face representation by combining instance-based representation and group-aware representation, GroupFace [74] has revealed that some visual properties reside in groups.

In this work, we interpret the face recognition model by designing a new framework called CondFace with dynamic convolutions [25] and then dive into an in-depth analysis of the intermediate representations. Contrary to standard convolution in which all samples share the same weights, in dynamic convolution [25], each kernel's parameters are a mixture of experts and an ensembling weight is calculated from the global view of the current input. For clarity, we describe the ensemble weight for each dynamic convolution as the routing weight and a set of routing weights from different positions as a routing vector. Empirically, we observe that the distribution of routing vectors captures facial appearance properties (Figure 5.1). However, this distribution is scattered and noisy. Based on these discoveries, we propose to improve from two aspects: (1) incorporating a clustering loss in training, (2) enhancing the representation ability of the routing vector. The introduced clustering loss will assign sample-dependent routing vectors into a fixed number of clusters. We can further analyse and interpret face recognition through facial attributes captured in the assigned clusters. To enhance routing vectors' representation and discrimination ability, we expand them with dynamic group convolution by incorporating a dynamic mechanism into group convolution. Thus each group convolution is equipped with individual experts. The dynamic group convolution is able to consider the split input.

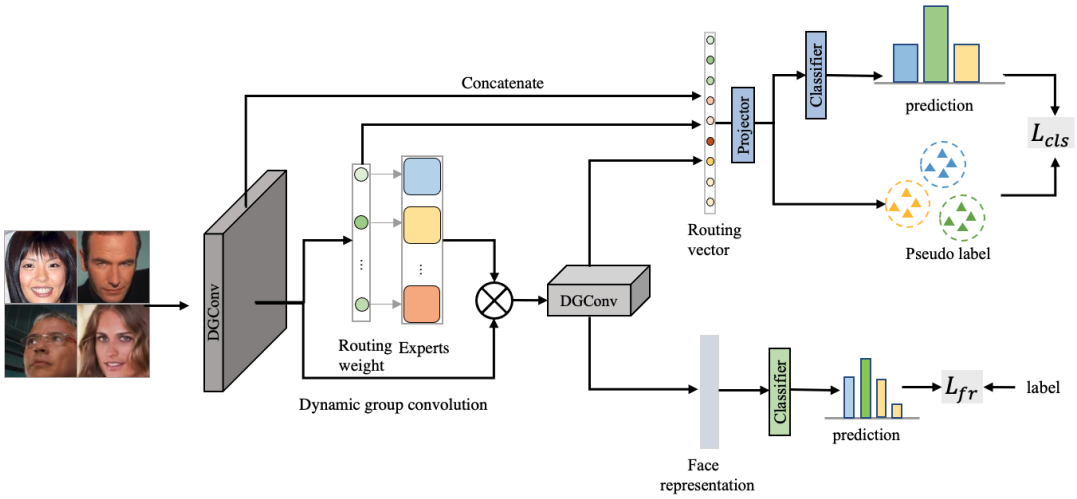


Fig. 5.2 Overview of the proposed framework. It combines a face recognition loss on the final feature representation with a clustering loss applied on intermediate routing weights extracted from several positions of the network. DGConv is short for Dynamic group convolution.

Our work is closely related to [185, 25] with the following distinctions. [185] is based on a random projection tree [31] and the leaf node is split based on the intermediate representation. However, our method replaces standard convolution with dynamic (conditional) convolution [25, 189] leading to a static network structure but with sample-specific continuous weights. Apart from this, in order to learn a tree-like network structure, an unsupervised constraint is imposed in [185]. The constraint is to partition data at the node by maximizing the distance between the centroids of two sub-clusters. As opposed to this, our method uses the unsupervised constraint to restrict the routing vectors' flexibility. For this purpose, we introduce a clustering loss to group the routing vectors of the entire dataset to a limited number of clusters. As a consequence, the routing vectors are now attribute-specific instead of merely being sample-specific. Beyond this, [74] qualitatively shows some group members without further investigation. We annotate 3 datasets with attributes like pose, gender, race, and age, and then carry out evaluation utilizing quantitative metrics from cluster analysis. The experimental results conspicuously show that clustering loss improves and concentrates routing vectors' distribution. We also offer adequate ablation studies on variants under the proposed framework. Moreover, we find an application, face inversion from the pre-trained network, and demonstrate that the routing vector contributes to faithful face reconstruction from its feature embedding.

The rest of this chapter is organized as follows. Section 5.2 describes the proposed method including two components: dynamic group convolution and unsupervised mixture of experts. Section 5.3 introduces the closely related work. Section 5.4 describes how to annotate faces with attributes, the metrics for evaluation, variants of the proposed method. Section 5.5 concludes this work.

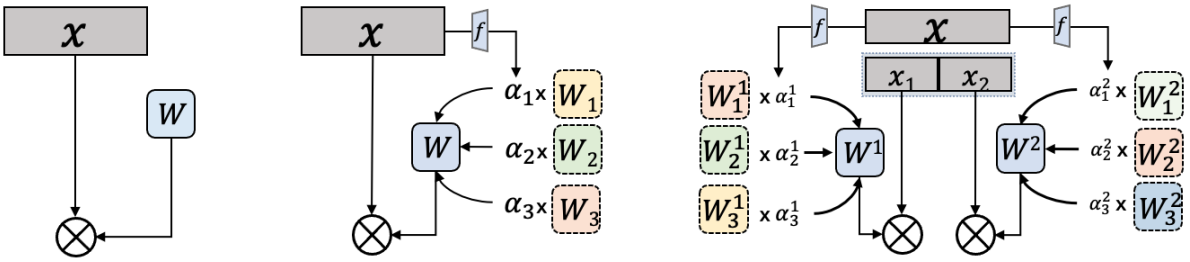


Fig. 5.3 Comparisons among three convolutions: (a) standard convolution, (b) dynamic convolution with expert number $n = 3$, (c) dynamic group convolution with groups $m = 2$ and expert number $n = 3$.

5.2 Method

Our goal is to learn visual concepts from training data without providing annotated attribute labels as supervision. The overview of the proposed method is in Figure 5.2. It consists of two losses: L_{cls} is the clustering loss on the routing vectors extracted from dynamic group convolutions at the mid-networks, and L_{fr} is the recognition loss on face representation from the network end. Unlike L_{fr} with supervision from the annotated label, L_{cls} is between the classifier's prediction and pseudo labels generated by clustering techniques. These two losses are trained together to constrain the routing vectors into limited clusters and also maintain the recognition performance. Therefore, we are able to interpret the model by analysing these assigned clusters. In Section 5.2.1, we introduce dynamic group convolution, and compare it with standard convolution and dynamic convolution. In Section 5.2.2, we describe the two clustering techniques to provide pseudo label for constraining the routing vectors. Finally, in Section 5.2.3, we integrate the clustering loss into the existing face recognition framework.

5.2.1 Dynamic Group Convolution

We first briefly formulate standard convolution and dynamic convolution, then move on to dynamic group convolution. For a regular convolution kernel weight W, the convolution operation ($\otimes$) with input x is formulated as:

$$O(x) = W \otimes x, \tag{5.1}$$

here W is shared by all training samples and optimized with the gradient from the task-based loss. $O(x)$ denotes x's output.

Unlike standard convolution, W in dynamic convolution [25] is sample-dependent. Given a filter basis pool with n experts $\{W_1, W_2, ..., W_n\}$, W is a mixture of the n experts with the routing weights ($\boldsymbol{\alpha} = \{\alpha_1, ...\alpha_n\}$) calculated on input x. The formulation is expressed as:

$$\begin{aligned} O(x) &= (\alpha_1 * W_1 + .. + \alpha_n * W_n) \otimes x \\ \boldsymbol{\alpha} &= f(x), \end{aligned} \tag{5.2}$$

where f represents the voted weight generator composed of a sequence of operations $AVG + LN + BN + ReLU + LN + Softmax$. Dynamic convolution [25] improves the flexibility and efficiency of the network without adding heavy computations in inference and shows superior performance on several tasks like image classification and keypoints detection. Thus, we apply this idea to face recognition task and empirically observe that the routing weights convey facial attributes like pose or gender. Encouraged by this behaviour, we go further to enhance its representation capacity by lengthening its dimension.

For this purpose, we instantiate dynamic group convolution. Basically, we divide W into m groups ($\{W^1, ..., W^k, ...W^m\}$) across input channel axis and then assign n experts ($\{W_1^k, ..., W_n^k\}$) to each group. We obtain single routing weight with dimension $m \times n$. The dynamic group convolution with input x is formulated as:

$$\begin{aligned} O(x) &= concat[W^1 \otimes x^1, W^k \otimes x^k, ..., W^m \otimes x^m] \\ W^k &= \alpha_1^k * W_1^k + ... + \alpha_n^k * W_n^k \\ \alpha^k &= f(x), \end{aligned} \tag{5.3}$$

where the input x is split into m groups $x \in \{x^1, x^2, ..., x^m\}$ on input channel axis. f generates the m group expert weights, each of which is a n dimensional vector. After that, each convolution weight W^k is assigned to a part of the input. Finally, the output of convolution layer is a combination of m segments. The visualization comparison among the three convolution is given in Figure 5.3.

5.2.2 Unsupervised Mixture of Experts

We define a cluster as a set of samples that share similar facial attributes. The cluster is determined by deploying online clustering techniques. We sequentially concatenate routing weights from K positions of the network to obtain the final routing vector $\boldsymbol{\beta}$ ($\beta = [\alpha_1, \alpha_2, ..., \alpha_K]$) and input to the clustering algorithms. Current methods [15, 207, 3, 16] combine clustering and representation learning together to capture class-level information. We introduce clustering techniques on the routing vectors and finally measure their capability to capture attribute-level information by analyzing the assigned clusters. The core idea of online deep clustering is to minimize the distance between the predicted label and the pseudo label generated from clustering techniques. The main difference among them is how to perform label assignment. Here, we choose two popular techniques: one is based on k-means on the whole dataset, and the other is based on Sinkhorn-Knopp [28] on the current batch.

k-means clustering: We follow DeepCluster [15] to do k-means clustering on the routing vectors. After each epoch, E_i, we perform k-means clustering on the extracted routing vectors on the training dataset. By fixing the cluster number as C, it divides the whole dataset into C groups and assigns a label to each sample. After adding a classifier, the goal of the next epoch, E_{i+1}, is to minimize the cross-entropy loss between the current prediction and the assigned pseudo label. Note that the k-means centroids will serve as the weight of the classifier.

Sinkhorn-Knopp: [16, 3] have replaced k-means in DeepCluster [15] with Sinkhorn-Knopp [28]. They firstly impose the constraints that all examples from each batch should be equally partitioned and pseudo labels for different images should be distinct. Label assignment is then equivalent to optimal transport solution which is solved efficiently by the iterative Sinhorn-Knopp algorithm [28]. Label assignment here is performed batch-wise for each iteration. Similarly, the network is trained to minimize the cross-entropy loss between current prediction from predictor and assigned label from Sinhorn-Knopp algorithm [28]. The prototypes in the classifier are optimized using clustering loss.

5.2.3 Learning

The interpretable face recognition model is trained simultaneously with a recognition loss and unsupervised clustering loss. The former is a margin-based softmax loss on the final representation to differentiate identities and the latter is a cross-entropy loss on the routing vectors to retrieve potential clusters.

Loss functions: The CosFace loss [162] L_{fr} to train face recognition task is described as:

$$L_{fr} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{\mathbf{y}_i})-m)}}{e^{s(\cos(\theta_{\mathbf{y}_i})-m)}+\sum_{j=1,j\neq\mathbf{y}_i}^{M}e^{s(\cos(\theta_j))}}, \tag{5.4}$$

where N denotes the batch size, and M is number of identities. θ_j is the angle between i feature embedding and class center j while y_i is i's associated class label. s is a scale factor, and m is a marginal factor. Arcface [34] can be also used here. We choose CosFace because Zhu *et al.* [228] have proved that when the margin in CosFace is increased to 0.48, its performance is comparable and some times even better than ArcFace [34].

Clustering loss reduces the cross entropy loss between current classifier's prediction $p(\boldsymbol{\beta}_i)$ and assigned pseudo label $P^*(\boldsymbol{\beta}_i)$ in Section 5.2.2. The formulation is defined as:

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}P^*(\boldsymbol{\beta}_i)\log(p(\boldsymbol{\beta}_i)). \tag{5.5}$$

Training: The whole network is optimized simultaneously with the above losses with parameter λ to balance the weight:

$$L = L_{fr} + \lambda L_{cls}. \tag{5.6}$$

5.3 Relationship to Previous Work

Dynamic deep neural networks. Unlike standard convolutional neural networks (CNNs) with fixed structure and weights for any input, dynamic CNNs [75, 66, 141, 30, 87, 165, 178, 221, 189, 25] produce different kernels, widths, or depths adapted to a given input. [75, 66, 141] have

fed sample's features into a function to learn individual kernels. Dai *et al.* [30] have learned convolutional offset for each example. [87, 165, 178] have learned sample-specific structures by skipping layers of an existing model. More recently, dynamic convolution with static network structure but sample-specific weights has gained much attention [221, 189, 25]. Yang *et al.* [189] have proposed conditional convolution by mixing the basis filters with sigmoid activated weights voted by input features. Concurrent work [25] has proposed key optimization tricks to ease the training difficulty using softmax with large temperature for kernel attention. Our work is inspired by [25] but we go further to enhance routing vectors' capacity by proposing dynamic group convolution. By operating on input segments, the final weight is composed of more basis filters. We introduce a clustering loss to guide the distribution of routing vectors and experimentally find that it is attribute related.

Unsupervised clustering: Recently, clustering-based unsupervised representation via jointly optimizing feature learning and image clustering has gained particular interest[15, 207, 3, 16]. The goal is to construct well-distinguished categories by preventing the assignment of most images to one or a few clusters. DeepCluster [15] suggests that k-means, considering relative metrics such as cosine similarity or Euclidean distance, effectively serves as label assignment to learn category-specific visual representations. Upon observing the phenomenon that DeepCluster leads to a degenerate solution where all data points mapped to the same cluster, Asano *et al.* [3] have added the equipartition label constraint with an optimal transport optimization. These ideas are further explored in [16], Caron *et al.* have mapped representations to prototypes and obtain simultaneous clustering and network update. Our work is closely related to the above. However, our input to clustering is the sample-specific weights, instead of the final representation itself. We learn the prototypes (centroids) as an intermediary to assign clusters, through which we explain the shared facial attributes residing in the networks.

Interpretable face recognition: There has been emerging research on the interpretation of face recognition models. Visualization of features from CNNs is the most straightforward solution and serves as a technical tool for diagnosing CNNs representations [187, 17, 205, 224, 172]. For example, [187] has visualized shape and texture determining the subject identity after training on a controlled dataset. [172] has generated a network attention map that best explains which regions in a probe image match a mated image.Other works attempt to explain what resides in the face template or an intermediate representation. Particularly, they use the performance of predicting facial attributes with features extracted from hidden layers to indicate how well the attribute is implicitly learned [113, 114, 112, 151, 37]. Our work seeks to explain the attribute information residing within the intermediate representation. However, we train our model without introducing explicit attribute labels. Further, we visualize the learned attributes by examining whether it will faithfully invert the input image via DeepInversion [199].

Fig. 5.4 Examples with annotated label '01111', '10410', '11122', '00222' respectively by each row. For example, '10410' denotes faces with attributes: age in 20-49, female, white, small pitch angle, and yaw angle '>30'.

5.4 Experiments

5.4.1 Datasets and Pre-processing

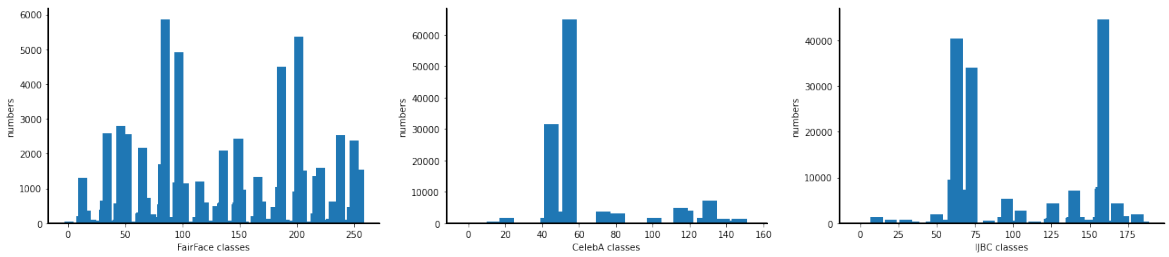
Training set: VGGFace2 [13] comprises 3.31M images of 9,131 subjects with large variations in pose, age, illumination, ethnicity and profession. We randomly selected 0.7M (roughly 80 images per identity) images as our training set. Note that we use the pre-processed version from ArcFace [34].

Evaluation set: We evaluate the models with two experiments: face recognition performance on the feature embedding and the clustering analysis on the intermediate representations, feature vectors and routing vectors. For the face verification experiments, we conduct evaluations on 5 benchmarks following the standard evaluation protocols.

- LFW [59] contains 13,233 images from 5,749 identities. It can be viewed as a milestone dataset in which images are crawled from the Internet containing variations in pose, illumination, expression, resolution, etc.
- CFPFP [136] contains 500 subjects, each has 10 frontal and 4 profile images.
- AgeDB [104] contains 12,240 images of 440 identities with age ranges in [1,101].
- IJB-B [171] contains 1,845 subjects (21.8K still images and 55K video frames). In total, there are 12,115 templates with 10,270 genuine matches and 8M impostor matches.
- IJB-C [100] is an extension of IJB-B [171], containing 3,531 subjects (31.3K still images and 117.5K video frames). In total, there are 23,124 templates with 19,557 genuine and 15,639K impostor matches.

To perform clustering analysis, we annotate each face image with 5 attributes (gender, age, race, yaw angle, pitch angle) with the pretrained models provided by FairFace [1] [70] and RetinaFace [35]. Then, each face is described as a 5 dimensional vector with sequential elements representing age ('0-19':0, '20-49':1, '50+':2), gender ('female':0, 'male':1), race ('Black':0, 'East Asian, Southeast Asian':1, 'Indian':2, 'Latino Hispanic':3, 'Middle Eastern, White':4), pitch ('<-30':0, '>30':1, 'others':2), yaw ('<-30':0, '>30':1, 'others':2). There are 270 cohorts in total but not all individuals are active (see Figure 5.5). We refer to Figure 5.4 with

[1] https://github.com/dchen236/FairFace



(a) FairFace (b) CelebA (c) IJBC

Fig. 5.5 Histogram distribution on defined labels across FairFace, CelebA and IJB-C. The label number simply denotes the ith class of the given dataset. Notice that not all labels have faces associated with in the current dataset, as the 5-D vector, defining a class, may represent a combination of properties that doesn't exist in the current data.

some examples. Further, we perform clustering analysis on the following 4 datasets FairFace[70], CelebA [96], IJB-C [100] and Sub200. Note that Sub200 is a combination dataset composed of 86 classes with 200 images per class after selecting images from the above 3 datasets. These 3 datasets in total has 86 classes with over 200 images. As is shown in Figure 5.5, all data distributions of FairFace, CelebA and IJBC are significantly imbalanced across defined labels (cohorts), thus we collect a balanced dataset.

- FairFace[70] contains 108,501 images labeled with race, gender, and age groups. We leverage RetinaFace [35] to predict pose (pitch and yaw).
- CelebA [96] is a large-scale face attributes (defined annotations like black hair, sunglasses, smiling) dataset with over 200,000 celebrity images. The data covers large pose variations and background clutter. We select images with confidence over 0.5 via FairFace [70] and then annotate this subset with RetinaFace [35]. In total, we collect 144,875 images.
- IJBC [100] is a popular evaluation dataset for face recognition task. We follow the procedures as in CelebA and in total collect 224,171 images.
- Sub200 is a combination dataset composed of 86 classes with 200 images per class after selecting images from the above 3 datasets. Note that these 3 datasets in total have 86 classes with over 200 images. As is shown in Figure 5.5, all data distributions of FairFace, CelebA and IJBC are significantly imbalanced across defined labels (cohorts), thus we collect a balanced dataset.

5.4.2 Evaluation Metrics

For verification: We compare the verification-accuracy for identity-pairs on LFW [59], CFP-FP [136], AgeDB [104]. We also compare a True Accept Rate (TAR) and False Accept Rate (FAR) $1e-4$ on IJB-B [171] and IJB-C [100] following the evaluation setting in ArcFace [34].

For clustering: We interpret the network by describing, analyzing and discovering implicit patterns in clusters. To validate clustering consistency with annotated labels, we measure the performance with the following two metrics:

- *Purity* evaluates inter-judge agreement in each cluster by calculating the rate between correctly assigned samples and cluster size. We define a majority class based on 4-5 attributes. A value close to 1.0 indicates a better match.

- *AMI* (Adjusted Mutual Information) is an improved version of NMI (Normalized Mutual Information) which is a balanced metric that can be used to determine the quality of clustering. Note that the limitation of NMI is that it does not have a constant 0 baseline value for two independent variables. A value close to 1.0 indicates a better match.

5.4.3 Implementation

We follow the publicly available code [2] for implementing and training our models. The baseline structure is IR-18 described in ArcFace [34]. Models are trained from scratch with an SGD optimizer, momentum of 0.9, weight decay of $5e-4$ and batch size of 512. The initial learning rate is 0.1 and reduced by a factor of 10 at epoch 16, 20, 24, 28 with total 32 epochs. The input face size is 112×112. The implementation of clustering methods is modified from public code [3]. The model with clustering loss is initialized from the pre-trained model to stabilize its training. The weight λ is set as 1.0. All models are implemented with PyTorch [119].

5.4.4 Variants and Analysis

For all experiments, we follow the training schedule and hyper-parameters mentioned earlier. To investigate the effects of individual components of proposed methods, we compare with two baseline models (BL, DC) and perform ablation studies on its variants (AS1-AS5).

BL is the baseline IR-18 network. In total, it has 4 layers with 2 blocks per layer and two convolutions per block.

DC denotes DC-IR-18 by replacing standard convolution with dynamic convolution with the expert number $n = 4$ in the last three layers of IR-18. It consists of 6 group routing weights in total after sharing the routing weight in a block. This specific setting of expert number and position is directly borrowed from [25] and we do not further investigate this as it already achieves superior performance (see Table 5.2).

AS1:number of prototypes. We train DC with clustering technique Sinkhorn-Knopp by setting prototype number as 30, 300, 1000 respectively and we get variants AS1:P=30, AS1:P=300, AS1:P=1000.

AS2:temperature in Softmax. Empirically, as temperature decreases, the output becomes more sparse. Following AS1, we set the prototype number as 300 with temperature as 0.1 and 15 respectively to get variants AS2:T=0.1, AS2:T=15.

[2] https://github.com/HuangYG123/CurricularFace

[3] https://github.com/facebookresearch/swav

Method	FairFace		CelebA		IJB-C		Sub200	
	Purity	AMI	Purity	AMI	Purity	AMI	Purity	AMI
BL:C=30	25.14	6.50	69.98	15.00	55.57	16.05	21.62	16.95
BL:C=300	34.27	11.71	77.19	18.11	70.52	24.21	44.34	24.37
BL:C=1000	39.75	13.04	70.00	18.28	78.71	29.62	56.88	21.57
DC:C=30	44.68	30.36	74.41	35.32	72.10	39.33	45.10	39.89
DC:C=300	52.11	27.25	77.21	29.48	79.42	37.33	54.47	38.09
DC:C=1000	55.25	23.42	86.71	25.96	82.81	35.88	62.64	28.82
AS1:P=30	49.62	28.89	77.04	36.49	73.24	40.02	46.31	40.87
AS1:P=300	59.49	29.82	84.57	33.02	79.81	39.03	55.31	39.11
AS1:P=1000	60.40	23.85	86.78	28.73	82.40	35.01	63.50	29.35
AS2:T=0.1	18.0	1.0	66.24	2.58	45.15	2.0	6.2	2.2
AS2:T=15	58.71	27.86	83.53	31.14	79.76	37.51	53.92	37.60
AS3:I.	63.60	31.08	87.0	**40.0**	82.91	41.18	62.40	39.21
AS3:L.	61.20	31.42	85.78	33.71	78.88	36.09	58.57	38.55
AS4	**65.53**	32.92	87.85	37.35	**85.91**	**43.22**	**65.16**	42.19
AS5	62.84	**34.59**	**88.65**	38.08	84.55	42.48	61.95	**44.46**

Table 5.1 Purity and AMI on FairFace, CelebA, IJB-C and Sub200. BL and DC are clustered with k-means. AS1-AS5 are clustered with trained classifier.

AS3:projection heads. With clustering loss on the routing vectors, we added a projection head to map routing vector and then appended a linear classifier with normalized weight to predict the label. As is mentioned in [44], the projection heads matter the final performance. Here, we mainly study 3 projection heads including (a) Identity mapping (I), (b) one Linear layer (L), and (c) multi-layer perceptron (MLP). We get variants AS1:P=300, AS3:I, AS3:L. Note that AS1:P=300 is trained with MLP by default. The prototype number in AS3 is 300.

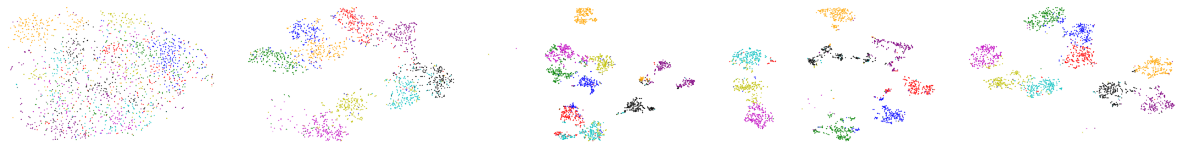
AS4:dynamic group convolution. The description is in Section 5.2.1. Specifically, we set the group number $m = 4$, the expert number $n = 4$, double the output channel number, and then squeeze the channel number back with a 1×1 convolution. This setting maintains the performance with similar parameters and flops with DC.

AS5:Different clustering techniques. Note that AS2-AS4 adopt Sinkhorn-Knopp to assign labels. In AS5, we replace Sinkhorn-Knopp with k-means on AS4.

Table 5.1 and Table 5.2 report clustering performance and face verification performance, respectively. Note that, in Table 5.1, the results for BL and DC are from offline k-means, and the rest are from trained classifier on routing vectors. Generally, a larger cluster (prototype) number consistently increases the purity. After balancing between purity and AMI, we choose cluster number as 300 and it is also the closest number to cohorts defined by attributes. Therefore, for AS2-AS5, the cluster (prototype) number is set as 300. The projection layer also matters but

Method	Verification Evaluation (%)				
	LFW	CFP-FP	Age-DB	IJB-B	IJB-C
BL	99.15	96.07	92.47	87.15	89.99
DC	99.23	96.62	93.45	88.27	90.73
AS1:P=30	99.32	96.61	93.32	88.33	90.92
AS1:P=300	99.32	**96.81**	93.57	88.20	90.85
AS1:P=1000	99.35	96.77	93.47	88.38	90.84
AS2:T=0.1	99.33	95.52	92.38	87.15	89.96
AS2:T=15	99.23	96.83	93.46	88.46	91.05
AS3:I	**99.36**	96.40	93.65	88.57	90.81
AS3:L	99.28	96.60	93.06	88.48	90.83
AS4	99.21	96.63	93.20	88.50	91.17
AS5	99.30	96.77	**93.75**	**88.80**	**91.55**

Table 5.2 Verification results on LFW, CFP-FP, AgeDB. 1:1 verification TAR (@FAR=1e-4) on the IJB-B and IJB-C.



(a) BL:C=300 (b) DC:C=300 (c) AS3:I (d) AS4 (e) AS5

Fig. 5.6 Feature visualization using t-SNE. The features are extracted by two baseline methods (BL:C=300,DC:C=300) and three proposed variants (AS3:I,AS4,AS5) respectively. For BL:C=300, features are extracted after last 3 layers. For the rest, features are the routing vectors. The images are from Sub200 dataset and different colors denote different labels.

the finding deviates from [16], here, the identity mapping alone achieves the best performance. Besides, we have the following findings:

Dynamic convolution boots recognition performance. As is shown in Table 5.2, DC outperforms BL by increasing $\sim 0.1\%$ on LFW, $\sim 0.6\%$ on CFPFP, $\sim 1\%$ on AgeDB, IJB-B, and IJB-C datasets. Besides, newly introduced clustering loss maintains its superior performance (see AS1 and AS3 in Table 5.2).

Routing vector reflects the facial attributes. We apply k-means with cluster number 30, 300, 1000 on layers' output from BL and routing vectors from DC respectively. To be specific, for BL, we first use the average pooling layer to transform the feature map from the last 3 layers to vector since their sizes vary from layer to layer and then concatenate them as $128+256+512=896$ vector as input to K-means. Table 5.1 summarizes the results. DC surpasses BL in all settings, which indicates routing weights' ability to capture the facial attributes significantly surpasses that of network activations (see Figure 5.6(a) and Figure 5.6(b)).

Fig. 5.7 Unsupervised discovery of facial attribute clustering from routing vectors. The dashed rectangle is the mean face of each the clusters and the right ones are examples belonging to each of them. The clusters are assigned by a trained classifier in AS5.

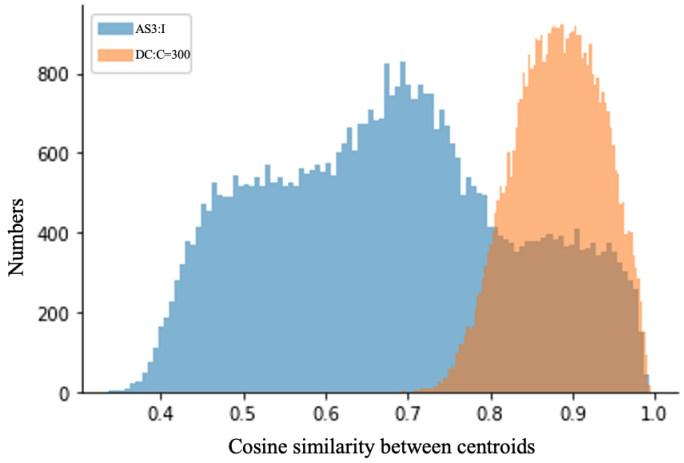


Fig. 5.8 Cosine distance distribution among the k-means centroids calculated from model DC:C=300 and AS3:I.

Clustering loss increases centroids' dissimilarity. Intuitively, the distance between centroids computed by k-means, not only serves as an inter-class similarity but also an indicator of the feature's discriminative capacity. Therefore, we follow the given steps to compare routing vectors from DC and AS1:I. Firstly, we calculate the routing vectors on Sub200 based on DC and AS1:I respectively. Then, we choose cluster number 300 for k-means on the individual routing features, assigning samples into 300 disjoint clusters. After obtaining the cluster centroids, we compute the cosine distance between each pair, and then plot histograms distribution by putting the similarity list into 100 bins. As we can see from Figure 5.8, the cluster centroids' similarity is significantly reduced when we introduce clustering loss in training, which intuitively indicates the effectiveness of the proposed training paradigm. Moreover, under this setting, the AMI for AS1 based on annotated labels is 39.59% which is 1.5% higher than DC with 38.09%.

Higher temperature will decrease the discrimination of routing vectors. For AS2, in Table 5.2, we find that too small temperature (0.1 in our case) value will damage the model performance as it will degrade into training with fewer routes. Besides, the higher temperature will not affect much on face recognition performance but it will cause the routing vector to soften and in turn affect its discriminative capacity.

Longer routing feature increase intra-class similarity. As is shown in Table 5.1, AS4 and AS5 with a 96-dimensional routing vector consistently increase purity and AMI across 4 datasets compared with AS1-AS3 with 24 dimensions. It is hard to decide the best candidate between AS4 and AS5 as in terms of purity, AS4 performs better while in terms of AMI, AS5 is better. However, it provides the evidence that popular clustering losses generalize well on this task. As shown in Figure 5.6, the added clustering loss helps increase the intra-class compactness (see Figure 5.6(b) and Figure 5.6(c)) and the longer routing weights increase inter-class discrimination (see Figure 5.6(c) and Figure 5.6(d)).

Figure 5.7 presents some examples of centroids of AS5 on Sub200 and assigned images in each cluster. Note that the learned clusters are highly associated with semantic concepts without attributes-related supervision. For example, the upper left centroid is white old men with profile faces.

We term AS5 as CondFace, which is a dynamic group convolution based structure trained with k-means clustering on its routing weights.

5.4.5 Face Inversion

This section demonstrates the capability of CondFace to effectively synthesize an input image in terms of attribute preservation and identity preservation. We adopt the network setting from AS5 and BL trained on MS1MV3 [46] to conduct the inversion experiments as experimentally we found that models trained on smaller dataset like VGGFace2 [13] generates unrealistic faces due to limited representation ability constrained by training dataset size. Models are trained from scratch with an SGD optimizer, momentum of 0.9, weight decay of $5e-4$ and batch size of 512. The initial learning rate is 0.1 and reduced by a factor of 10 at epoch 10, 18, 22 with total 24 epochs. The training dataset is MS1MV3 [46]. The weight λ is set as 0.1. All models are implemented with PyTorch [119].

We follow public code [4] of DeepInversion [199]. We synthesize 112×112 face images with batch size 192 using one NVIDIA V100 GPU. All models are optimized using Adam optimizer with cosine learning rate decay at an initial learning rate 0.25 and optimized 20K steps in total. Regularization parameters for total variance and ℓ_2 norm loss are set as $1e-3$ and $1e-4$, respectively. Both identity preservation and routing vector preservation are constrained by a ℓ_2 loss with weights 1.0 and 10.0, respectively.

To quantitatively validate the quality of the inverted images, three metrics are adopted: (1) structural similarity (SSIM);(2) cosine similarity from ResNet101 trained on MS1MV3; and (3) face verification accuracy on LFW with ResNet101 trained on MS1MV3). Comparison is reported in Table 5.3. We observe that CondFace improves performance considerably compared to BL. It is worth mentioning that the performance of ResNet101 drops 9% on LFW inverted from BL while only decreases by 2% on that from CondFace. It confirms the effectiveness of CondFace in image reconstruction. Considering the image value is constrained by batch normalization

[4] https://github.com/NVlabs/DeepInversion

Method	SSIM	Cosine Sim	LFW Acc (%)
BL	0.21	0.57	90.09
CondFace	**0.30**	**0.72**	**97.79**

Table 5.3 Structural similarity, cosine similarity and verification accuracy on LFW of baseline model and CondFace. The cosine similarity and the verification accuracy are evaluated with ResNet101 trained on MS1MV3. Acc for BL, CondFace, ResNet101 on original LFW are 99.62%, 99.73%, 99.82% respectively.

Fig. 5.9 Face inversion results. The first row is the input image. The second row shows inversion results from BL and the third row shows inversion results from CondFace. The left 6 columns are male faces across different ages, faces, and poses while the rest 6 columns are for female.

priors, it is quite understandable that SSIM is not high but CondFace still surpass BL by 0.1%. Also, the cosine similarity obviously increases from 0.57 to 0.72. Visualization examples are illustrated in Figure 5.9. We argue routing vectors reduce the ambiguity in generating faces affected by attributes, especially the pose prior, and therefore produce faithful images. For instance, the poses are maintained quite well for CondFace while missed in BL in Figure 5.9.

5.5 Conclusion

The main idea of this work is to learn visual attributes with a supervised face recognition loss and an unsupervised clustering loss. We are the first to analyse and interpret face recognition through routing vectors in dynamic convolution. This is motivated by the dynamic convolution's behaviour in attribute learning. However, we observed the gained distribution is noisy and scattered. Therefore, we propose to leverage an unsupervised clustering loss to push the routing vectors to a fixed number of clusters in the feature space without damaging the recognition performance. Also, we incorporate dynamic convolution into group convolution by considering the piece-by-piece input prior to enhance routing vectors' representative ability. In addition, due to the lack of annotated datasets with desired facial properties, we annotate 3 datasets with attributes and compare baseline methods and variants both quantitatively and qualitatively on clustering quality. The experiments verify the effectiveness and efficiency of the proposed framework. Finally, we also confirm that the routing vector benefits the feature-based face reconstruction via DeepInversion both visually and quantitatively. The comparison have shown that image quality with routing vectors is more faithful to the original input. The proposed method

can be integrated in the “FAN-Face" to interpret its behavior. We argue similar conclusion can be reached as the inner weights are focusing on facial attributes while the final face embedding encodes the identification. Besides, once the knowledge distillation in Chapter 4 is explored, we assume the size of CondFace can be reduced without sacrificing its interpretability.

Chapter 6

Conclusions

6.1 Summary of Thesis Achievements

In this thesis, we have studied the face recognition problem from three aspects including using face alignment prior to improve face recognition performance by establishing shape correspondence, improving low capacity student network's performance with the high-capacity teacher network by enforcing the similarity of penultimate layer, further boosting student's performance with extra unlabeled data, and interpreting the behaviour of face recognition model by conducting clustering analysis on inner network parameters.

In **Chapter3**, we have proposed "FAN-Face" which uses features from a pretrained facial landmark localization network as a location prior to enhance unaligned face recognition performance. Motivated by the situation that facial landmarks provide pose, expression and shape information, therefore, we investigate to use knowledge of these landmarks for establishing correspondence among faces by establishing the shape correspondence. Contrary to most current algorithms which use facial landmarks to do face cropping in order to remove scale, rotation and translation variations, we propose a simple approach to face recognition which gradually integrates features from different layers of a facial landmark localization network into different layers of the recognition network. To the best of our knowledge, we are the first to explore features from a pretrained facial landmark localization model to enhance face recognition accuracy. In particular, both landmark heatmaps and features from the facial landmark network are integrated into the face recognition feature extraction process to (a) provide facial pose-, expression-, and shape-related information, and (b) help establish correspondence for improving face matching. Besides, we explore various architectural design choices at the network level to identify the best strategy for integration. The key design is a novel feature integration layer that is able to effectively integrate the features from the two networks although they are trained with very different objectives and loss functions. Through extensive experiments on several face recognition benchmarks including LFW [59], YTF [173], IJB-B [171], IJB-C [100] and MegaFace [71], we have illustrated how the proposed approach, when integrated with existing state-of-the-art methods, systematically improves face recognition accuracy for a wide variety of experimental settings and set a new state-of-the-art on these challenging datasets.

In **Chapter4**, we address the problem of model compression via knowledge distillation by advocating for a method that optimizes the output feature of the penultimate layer of the student network. The advantage is to directly optimize representation learning. Motivated by the success of FitNet [129], we firstly propose a direct feature matching approach that focuses on optimizing the student's penultimate layer only. However, as feature matching does not take into account the classification problem at hand by treating all channels in the feature vector equally, we propose to further decouple representation learning and classification. Specifically, we utilize the teacher's pre-trained classifier to train the student's penultimate layer feature to weight channel representation. In particular, for the same input image, we wish the teacher's and student's feature to produce the same output when passed through the teacher's classifier, which is achieved with a simple L_2 loss. The ablation studies on the variants of proposed representation learning have validated the effectiveness of our method in enforcing teacher-student similarity, reducing representation distance, improving the quality of feature clustering and boosting transferability of representation. Another contribution is that we expand the connection of our method with representation learning and show that it is possible to exploit unlabeled, incurred data to expand the support used to transfer from the teacher to the student, further boosting the student's performance. The rationale behind the knowledge distillation loss on unlabeled data is that the teacher network trained on labeled datasets is capable of capturing rich relational information among labels both in class space and feature space. Our method is extremely simple to implement and straightforward to train and is shown to consistently outperform previous state-of-the-art methods over a large set of experimental settings including different (a) network architectures, (b) teacher-student capacities, (c) datasets, (d) domains, (e) tasks. We argue the proposed method can be employed to enhance low capacity models' performance in "FAN-Face" by two intuitive ways. One is to train a high capacity teacher to guide the training of low capacity student and both the networks are equipped with FAN model. This is a simple instantiation of teacher-present knowledge distillation. The other is to only assist teacher with FAN model in training while the student without FAN knowledge is to mimic teacher's feature representation. By doing so, student is supposed to learn the shape correspondence from "FAN-Face" based teacher and we can get a light weight student without FAN model with descent performance.

In **Chapter5**, we propose to learn visual attributes with a supervised face recognition loss at the end of the feature extraction network and an unsupervised clustering loss at the middle of inner parameters. This is motivated by the dynamic convolution's behaviour in attribute learning. To the best of our knowledge, we are the first to analyse and interpret face recognition through routing vectors in dynamic convolution. Also, to enhance routing vectors' representative ability, we propose to incorporate dynamic convolution into group convolution in order to consider the piece-by-piece input prior. However, we observed that the routing vector in dynamic convolution is capable of partitioning data based on representational attributes but results in a noisy and scattered distribution. Therefore, we propose to leverage an unsupervised clustering loss to push the routing vectors to a fixed number of clusters in the feature space without damaging the recognition performance. Due to the lack of annotated datasets with desired facial properties, we

annotate three datasets with attributes like gender, age, race, pose. We further compare baseline methods and variants both quantitatively and qualitatively on clustering quality. The experiments have validated the effectiveness and efficiency of the proposed framework. Finally, we also confirm that the routing vector benefits the feature-based face reconstruction via DeepInversion and both visually and quantitatively comparison have shown that image quality with routing vectors is more faithful to the original input. This paradigm can also be extended to “FAN-Face" structure under the unaligned setting. Same conclusion of what is resided in inner weight may be achieved. Also, the knowledge distillation proposed in Chapter 4 can be used to reduce the model size by decreasing the channel number per convolution layers but maintaining the interpretability.

6.2 Future Directions

There are plenty of possible directions to explore to extend the research in this thesis.

Data-free face recognition. The success of deep face recognition heavily relies on a large amount of annotated data. However, it is privacy-sensitive to release these datasets online and some large datasets like MegaFace have already been withdrawn from the website. Also, some social media companies have internal datasets to gain high performance but they will not release them due to lots of concerns. Therefore, it is a problem worth exploring to invert the model feature to its corresponding face image on the public model. This technique called deep inversion has been studied in general object classification task but has not been explored in face recognition task due to its time-consuming optimization. For instance, it takes me around 2 days on 4 V100 GPUS to invert the LFW datasets on the ResNet18 model. It would take several months to invert whole MS1M datasets. As is found in Chapter 5, the attributes help to improve image quality. It also accelerates the inversion as the provided attributes reduce the ambiguity in generating a face. If this problem can be solved, we are able to dissect the datasets from the pre-trained model then use it to train our model. Therefore, face inversion is a promising topic to link industry and academy in the face recognition field.

Self-supervised face recognition. Recently, self-supervised learning under contrastive learning framework gains much attention and has boosted accuracy on ImageNet after improving the feature representation ability. The proxy task is data augmentation related and targets to narrow the distance of input after different augmentations. However, so far, few works have explored this idea in face recognition. The face recognition datasets are align cropped, and only randomly flipped augmentation is utilized. Although in our “FAN-Face" work, unaligned face is used instead of aligned version but it is still tightly cropped with less flexibility. Therefore, improving unaligned loose cropped face recognition performance is a promising topic and self-supervised learning is beneficial to obtaining augmentation robust face embedding.

Face recognition model training on Long-tail, Noisy and Biased Data. Most currently widely used training datasets like MS1M, VGGFace2, CASIA, are both breath (large ID numbers) and depth abundant (sufficient number per ID). To further improve the performance, downloading the face images based on a pre-defined celebrity list to create a massive dataset is a direct approach.

However, the final obtained dataset is always limited in depth. For example, there only two face images for some IDs. This situation is defined as shallow face learning and current face recognition losses struggle to train well due to lack of intra-class diversity in these IDs. Therefore, how to select adaptive margin in margin-based softmax loss on shallow training datasets is an interesting direction.

References

[1] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai. Variational information distillation for knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 62, 63

[2] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Unsupervised label noise modeling and loss correction. *International Conference on Machine Learning*, 2019. 51

[3] Y. Asano, C. Rupprecht, and A. Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. 73, 74, 75

[4] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *IEEE international joint conference on biometrics*, 2017. 67

[5] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 23, 24

[6] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances on Neural Information Processing Systems*, 2019. 52

[7] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006. 45

[8] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *IEEE International Conference on Computer Vision*, 2017. 12, 28

[9] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *IEEE International Conference*

on Computer Vision, 2017. 28

[10] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *IEEE International Conference on Computer Vision*, 2017. 12

[11] A. Bulat and G. Tzimiropoulos. XNOR-Net++: Improved binary neural networks. In *British Machine Vision Conference*, 2019. 59

[12] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy. Pose-robust face recognition via deep residual equivariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 69

[13] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *IEEE international conference on automatic face & gesture recognition*, 2018. 2, 34, 39, 40, 67, 76, 82

[14] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal on Computer Vision*, 2014. 11

[15] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018. 5, 73, 74, 75

[16] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances on Neural Information Processing Systems*, 2020. 73, 74, 75, 80

[17] G. Castanon and J. Byrne. Visualizing and quantifying discriminative features for face recognition. In *IEEE international conference on automatic face & gesture recognition*, 2018. 75

[18] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, 2018. 21, 22, 23

[19] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. In *Advances on Neural Information Processing Systems*, 2019. 23, 24, 25

[20] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In *European Conference on Computer Vision*, 2016. 8, 9

[21] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances on Neural Information Processing*

Systems, 2017. 15, 16

[22] R. Chen, H. Chen, J. Ren, G. Huang, and Q. Zhang. Explaining neural networks semantically and quantitatively. In *IEEE International Conference on Computer Vision*, 2019. 25

[23] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, 2018. xii, 59, 60, 67

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 46

[25] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu. Dynamic convolution: Attention over convolution kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 70, 71, 72, 73, 74, 75, 78

[26] J. H. Cho and B. Hariharan. On the efficacy of knowledge distillation. In *IEEE International Conference on Computer Vision*, 2019. 15, 16

[27] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011. 55

[28] M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances on Neural Information Processing Systems*, 2013. 5, 73, 74

[29] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances on Neural Information Processing Systems*, 2016. 9

[30] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision*, 2017. 74, 75

[31] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *ACM symposium on Theory of computing*, 2008. 71

[32] J. Deng, S. Cheng, N. Xue, Y. Zhou, and S. Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 33

[33] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *European Conference on Computer Vision*, 2020. 15

[34] J. Deng, J. Guo, X. Niannan, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. viii, 2, 3, 15, 27, 28, 33, 34, 35, 38, 39, 40, 41, 42, 43, 59, 67, 74, 76, 77, 78

[35] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 8, 10, 76, 77

[36] J. Deng, Q. Liu, J. Yang, and D. Tao. M3 csr: Multi-view, multi-scale and multi-component cascade shape regression. *Image and Vision Computing*, 2016. 11

[37] P. Dhar, A. Bansal, C. D. Castillo, J. Gleason, P. J. Phillips, and R. Chellappa. How are attributes expressed in face dcnns? *arXiv*, 2019. 70, 75

[38] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 23

[39] S. S. Du and J. D. Lee. On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, 2018. 45

[40] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 2009. 21

[41] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 11, 12, 37

[42] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015. 9

[43] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari. Do semantic parts emerge in convolutional neural networks? *International Journal on Computer Vision*, 2018. 23, 24

[44] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances on Neural Information Processing Systems*, 2020. 79

[45] Y. Guan, P. Zhao, B. Wang, Y. Zhang, C. Yao, K. Bian, and J. Tang. Differentiable feature aggregation search for knowledge distillation. In *European Conference on Computer Vision*, 2020. 18

[46] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 2016. 2, 3, 82

[47] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 2016. 34, 59

[48] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 3, 45

[49] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 46

[50] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, 2017. 9

[51] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 28

[52] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 56

[53] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi. A comprehensive overhaul of feature distillation. In *IEEE International Conference on Computer Vision*, 2019. 15, 18, 56, 57, 58, 59

[54] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI Conference on Artificial Intelligence*, 2019. 18, 62, 63

[55] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 15, 16, 45, 46, 47, 48, 49, 50, 54, 56, 57, 58, 59, 60, 61, 62, 63, 65, 67

[56] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning lightweight lane detection cnns by self attention distillation. In *IEEE International Conference on Computer Vision*, 2019. 20, 21

[57] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 56, 58

[58] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense networks for resource efficient image classification. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017. 20

[59] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008. 3, 4, 26, 28, 34, 38, 42, 68, 76, 77, 85

[60] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision*, 2017. 53

[61] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[62] Z. Huang and N. Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017. 18, 62, 63

[63] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Label propagation for deep semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 64

[64] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *IEEE International Conference on Computer Vision*, 2017. 30

[65] H. Jain, S. Gidaris, N. Komodakis, P. Pérez, and M. Cord. Quest: Quantized embedding space for transferring knowledge. In *European Conference on Computer Vision*, 2020. 19

[66] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *Advances on Neural Information Processing Systems*, 2016. 74

[67] H. Jiang and E. Learned-Miller. Face detection with the faster r-cnn. In *IEEE international conference on automatic face & gesture recognition*, 2017. 9

[68] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *International Conference on Machine Learning*, 2018. 51

[69] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. 46

[70] K. Karkkainen and J. Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *IEEE Winter Conference on Applications of Computer Vision*, 2021. 76, 77

[71] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4, 26, 28, 34, 38, 41, 85

[72] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 59

[73] J. Kim, S. Park, and N. Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances on Neural Information Processing Systems*, 2018. 62, 63

[74] Y. Kim, W. Park, M.-C. Roh, and J. Shin. Groupface: Learning latent groups and constructing group-based representations for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 70, 71

[75] B. Klein, L. Wolf, and Y. Afek. A dynamic convolutional layer for short range weather prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 74

[76] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. In *Technical Report*, 2009. 57, 61

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances on Neural Information Processing Systems*, 2012. 18, 43

[78] A. Kumar, T. K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 12

[79] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations*, 2017. 51

[80] V. Lebedev and V. Lempitsky. Fast convnets using group-wise brain damage. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 45

[81] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning Workshop*, 2013. 51, 64

[82] S. H. Lee, D. H. Kim, and B. C. Song. Self-supervised knowledge distillation using singular value decomposition. In *European Conference on Computer Vision*, 2018. 15, 19

[83] C. Li, J. Peng, L. Yuan, G. Wang, X. Liang, L. Lin, and X. Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 18

[84] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8

[85] O. Li, H. Liu, C. Chen, and C. Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. *AAAI Conference on Artificial Intelligence*, 2018. 24, 25

[86] X. Li, J. Wu, H. Fang, Y. Liao, F. Wang, and C. Qian. Local correlation consistency for knowledge distillation. In *European Conference on Computer Vision*, 2020. 19

[87] J. Lin, Y. Rao, J. Lu, and J. Zhou. Runtime neural pruning. In *Advances on Neural Information Processing Systems*, 2017. 74, 75

[88] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017. 9

[89] B. Liu, W. Deng, Y. Zhong, M. Wang, J. Hu, X. Tao, and Y. Huang. Fair loss: margin-aware reinforcement learning for deep face recognition. In *IEEE International Conference on Computer Vision*, 2019. 15

[90] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1611.01578*, 2018. 3, 45

[91] H. Liu, X. Zhu, Z. Lei, and S. Z. Li. Adaptiveface: Adaptive margin and sampling for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 15

[92] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015. 13, 14

[93] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016. 9

[94] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 15, 27, 33, 34, 41, 43, 67

[95] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan. Knowledge distillation via instance relationship graph. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 19

[96] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015. 77

[97] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 21

[98] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval (chapter 16)*. Cambridge university press, 2008. 55

[99] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 33

[100] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, et al. Iarpa janus benchmark–c: Face dataset and protocol. In

International Conference on Biometrics, 2018. 3, 4, 26, 28, 34, 38, 76, 77, 85

[101] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2018. 51

[102] H. Mobahi, M. Farajtabar, and P. L. Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances on Neural Information Processing Systems*, 2020. 21

[103] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. *google AI blog*, 2015. 21, 22

[104] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2017. 3, 68, 76, 77

[105] R. Naidu and J. Michael. Ss-cam: Smoothed score-cam for sharper visual feature localization. *arXiv*, 2020. 22, 23

[106] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *IEEE International Conference on Computer Vision*, 2017. 8, 10

[107] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, 2016. 28

[108] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *IEEE International Conference on Image Processing*, 2014. 41

[109] A. Nguyen, J. Yosinski, and J. Clune. Understanding neural networks via feature visualization: A survey. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019. 21

[110] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. 21, 22

[111] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv*, 2019. 22, 23

[112] A. J. O'Toole, C. D. Castillo, C. J. Parde, M. Q. Hill, and R. Chellappa. Face space representations in deep convolutional neural networks. *Trends in cognitive sciences*, 2018. 70, 75

[113] C. J. Parde, C. Castillo, M. Q. Hill, Y. I. Colon, S. Sankaranarayanan, J.-C. Chen, and A. J. O'Toole. Face and image representation in deep cnn features. In *IEEE international conference on automatic face & gesture recognition*, 2017. 70, 75

[114] C. J. Parde, Y. Hu, C. Castillo, S. Sankaranarayanan, and A. J. O'Toole. Social trait information in deep convolutional neural networks trained for face identification. *Cognitive science*, 2019. 70, 75

[115] W. Park, D. Kim, Y. Lu, and M. Cho. Relational knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 15, 19, 56, 57, 58, 59, 60, 61, 62, 63

[116] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *British Machine Vision Conference*, 2015. 13, 14, 34, 67

[117] N. Passalis and A. Tefas. Learning deep representations with probabilistic knowledge transfer. In *European Conference on Computer Vision*, 2018. 60, 61, 62, 63

[118] N. Passalis, M. Tzelepi, and A. Tefas. Heterogeneous knowledge distillation using information flow modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 18

[119] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *pytorch*, 2017. 34, 58, 78, 82

[120] B. Peng, X. Jin, J. Liu, S. Zhou, Y. Wu, Y. Liu, D. Li, and Z. Zhang. Correlation congruence for knowledge distillation. In *IEEE International Conference on Computer Vision*, 2019. 19, 62, 63

[121] H. Pham, Q. Xie, Z. Dai, and Q. V. Le. Meta pseudo labels. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 65

[122] M. Phuong and C. H. Lampert. Distillation-based training for multi-exit architectures. In *IEEE International Conference on Computer Vision*, 2019. 20

[123] H. G. Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *IEEE Winter Conference on Applications of Computer Vision*, 2020. 22, 23

[124] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *IEEE international conference on automatic face & gesture recognition*, 2017. 69

[125] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016. 3, 45, 59

[126] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 11

[127] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2016. 9

[128] E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003. 51

[129] 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. 15, 17, 45, 46, 47, 62, 63, 86

[130] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal on Computer Vision*, 2015. 57, 58, 62, 65

[131] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. *Advances on Neural Information Processing Systems*, 2017. 24, 25

[132] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances on Neural Information Processing Systems*, 2016. 51

[133] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 56

[134] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 13, 14, 33, 51

[135] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, 2017. 21, 22, 23

[136] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, 2016. 3, 34, 38, 42, 68, 76, 77

[137] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 24

[138] M. Simon, E. Rodner, and J. Denzler. Part detector discovery in deep convolutional neural networks. In *Asian Conference on Computer Vision*, 2014. 23, 24

[139] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv*, 2013. 21, 23

[140] M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 2018. 45

[141] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz. Pixel-adaptive convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 74

[142] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 37

[143] X. Sun, P. Wu, and S. C. Hoi. Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing*, 2018. 9

[144] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv*, 2015. 2, 13

[145] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 11

[146] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2, 13

[147] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 13

[148] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 33

[149] Z. Tang, X. Peng, S. Geng, L. Wu, S. Zhang, and D. Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *European Conference on Computer Vision*, 2018. 12

[150] Z. Tang, X. Peng, K. Li, and D. N. Metaxas. Towards efficient u-nets: A coupled and quantized approach. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 2019. 12

[151] P. Terhörst, D. Fährmann, N. Damer, F. Kirchbuchner, and A. Kuijper. Beyond identity: What information is stored in biometric face templates? In *IEEE international joint conference on biometrics*, 2020. 70, 75

[152] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016. 65

[153] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020. 19, 50, 55, 56, 58, 59, 62, 63, 65, 67

[154] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 33

[155] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 11

[156] F. Tung and G. Mori. Similarity-preserving knowledge distillation. In *IEEE International Conference on Computer Vision*, 2019. 19, 56, 62, 63

[157] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. *International Conference on Machine Learning*, 2018. 52

[158] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019. 52

[159] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy. The devil of face recognition is in the noise. In *European Conference on Computer Vision*, 2018. 15

[160] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 2017. 2, 14

[161] H. Wang, Z. Li, X. Ji, and Y. Wang. Face r-cnn. *arXiv*, 2017. 9

[162] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 15, 27, 33, 34, 38, 41, 43, 67, 74

[163] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2020. 22

[164] X. Wang, L. Bo, and L. Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *IEEE International Conference on Computer Vision*, 2019. 12

[165] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, and J. E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, 2018. 74, 75

[166] Y. Wang, H. Su, B. Zhang, and X. Hu. Interpret neural networks by identifying critical data routing paths. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 24, 25

[167] Z. Wang, K. He, Y. Fu, R. Feng, Y.-G. Jiang, and X. Xue. Multi-task deep neural network for joint face recognition and facial attribute prediction. In *ACM on International Conference on Multimedia Retrieval*, 2017. 69

[168] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 29

[169] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 2009. 13, 51

[170] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, 2016. 27, 33, 34, 41, 67

[171] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, et al. Iarpa janus benchmark-b face dataset. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2017. 3, 4, 26, 28, 34, 38, 76, 77, 85

[172] J. R. Williford, B. B. May, and J. Byrne. Explainable face recognition. In *European Conference on Computer Vision*, 2020. 75

[173] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 4, 26, 34, 38, 42, 85

[174] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 45

[175] W. Wu, M. Kan, X. Liu, Y. Yang, S. Shan, and X. Chen. Recursive spatial transformer (rest) for alignment-free face recognition. In *IEEE International Conference on Computer Vision*, 2017. 43

[176] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 12, 60

[177] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 2018. 34, 59

[178] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris. Blockdrop: Dynamic inference paths in residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 74, 75

[179] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 50

[180] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, 2018. 37

[181] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 51

[182] W. Xie, L. Shen, and A. Zisserman. Comparator networks. In *European Conference on Computer Vision*, 2018. 39, 40

[183] W. Xie and A. Zisserman. Multicolumn networks for face recognition. *British Machine Vision Conference*, 2018. 39, 40

[184] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances on Neural Information Processing Systems*, 2003. 13, 51

[185] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim. Conditional convolutional neural network for modality-aware face recognition. In *IEEE International Conference on Computer Vision*, 2015. 70, 71

[186] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 11

[187] T. Xu, J. Zhan, O. G. Garrod, P. H. Torr, S.-C. Zhu, R. A. Ince, and P. G. Schyns. Deeper interpretability of deep networks. *arXiv*, 2018. 75

[188] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 65

[189] B. Yang, G. Bender, Q. V. Le, and J. Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances on Neural Information Processing Systems*, 2019. 71, 74, 75

[190] C. Yang, L. Xie, C. Su, and A. L. Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 21

[191] J. Yang, A. Bulat, and G. Tzimiropoulos. Fan-face: a simple orthogonal improvement to deep face recognition. In *AAAI Conference on Artificial Intelligence*, 2020. 26

[192] J. Yang, J. Deng, K. Zhang, and Q. Liu. Facial shape tracking via spatio-temporal cascade shape regression. In *IEEE International Conference on Computer Vision Workshop*, 2015. 11

[193] J. Yang, Q. Liu, and K. Zhang. Stacked hourglass network for robust facial landmark localisation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2017. 12

[194] J. Yang, B. Marinez, A. Bulat, and G. Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2021. 44

[195] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 2, 34, 41, 42

[196] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 13

[197] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 15, 18

[198] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 22

[199] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 75, 82

[200] X. Yin and X. Liu. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 2018. 36, 37

[201] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *IEEE International Conference on Computer Vision*, 2017. 33

[202] Y. Yoo, D. Han, and S. Yun. Extd: Extremely tiny face detector via iterative filter reuse. *arXiv*, 2019. 10

[203] S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016. 56

[204] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. 3, 15, 17, 18, 20, 45, 46, 47, 49, 50, 54, 56, 57, 58, 59, 60, 61, 62, 63

[205] T. Zee, G. Gali, and I. Nwogu. Enhancing human face recognition with an interpretable neural network. In *IEEE International Conference on Computer Vision Workshop*, 2019. 75

[206] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014. 21, 23

[207] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy. Online deep clustering for unsupervised representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 73, 75

[208] F. Zhang, X. Zhu, and M. Ye. Fast human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 16

[209] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018. 50, 52

[210] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016. 8

[211] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu. Detecting faces using inside cascaded contextual cnn. In *IEEE International Conference on Computer Vision*, 2017. 8

[212] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 20

[213] L. Zhang, Z. Tan, J. Song, J. Chen, C. Bao, and K. Ma. Scan: A scalable neural networks framework towards compact and efficient models. *Advances on Neural Information Processing Systems*, 2019. 20

[214] Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu. Interpreting cnn knowledge via an explanatory graph. In *AAAI Conference on Artificial Intelligence*, 2018. 24

[215] Q. Zhang, R. Cao, Y. N. Wu, and S.-C. Zhu. Growing interpretable part graphs on convnets via multi-shot learning. In *AAAI Conference on Artificial Intelligence*, 2017. 24

[216] Q. Zhang, Y. Nian Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 23, 24

[217] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 24

[218] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Faceboxes: A cpu real-time face detector with high accuracy. In *IEEE international joint conference on biometrics*, 2017. 10

[219] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S3fd: Single shot scale-invariant face detector. In *IEEE International Conference on Computer Vision*, 2017. 8, 9

[220] Z. Zhang and M. R. Sabuncu. Self-distillation as instance-specific label smoothing. *Advances on Neural Information Processing Systems*, 2020. 21

[221] F. Zhao, J. Zhao, S. Yan, and J. Feng. Dynamic conditional networks for few-shot learning. In *European Conference on Computer Vision*, 2018. 74, 75

[222] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 2018. 68

[223] T. Zheng, W. Deng, and J. Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017. 68

[224] Y. Zhong and W. Deng. Exploring features and attributes in deep face recognition using visualization techniques. In *IEEE international conference on automatic face & gesture recognition*, 2019. 75

[225] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations*, 2015. 23

[226] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 22, 23

[227] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In *Deep learning for biometrics*, 2017. 9

[228] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du, and J. Zhou. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *CVPR*, 2021. 74

[229] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv*, 2016. 3, 45