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

Jing Yang,[1,*] **Adrian Bulat,**[2] **Georgios Tzimiropoulos**[1,2]
[1]University of Nottingham, [2]Samsung AI Center, Cambridge
{jing.yang2, yorgos.tzimiropoulos}@nottingham.ac.uk, adrian@adrianbulat.com

Abstract

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. This paper proposes 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 most difficult face recognition datasets, setting a new state-of-the-art on IJB-B, IJB-C and MegaFace datasets.

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 to all existing state-of-the-art face recognition methods.

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, Convolutional Neural Networks (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 (Wen et al. 2016; Liu et al. 2017; Wang et al. 2018; Deng et al. 2019) 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 which proposes to do so.

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 which is able to effectively integrate the features from the

* Work done as an intern at Samsung AI Center, Cambridge

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 IJB-B (Whitelam et al. 2017), IJB-C (Maze et al. 2018) and MegaFace (Kemelmacher-Shlizerman et al. 2016) datasets.

2 Related Work

There is a long list of deep learning papers for face recognition, and a detailed review of this topic is out of scope of this section. Herein, we focus only on two lines of work that have been shown to improve accuracy especially across large pose variations.

New loss functions. The first line of work includes a number of papers (Schroff, Kalenichenko, and Philbin 2015; Wen et al. 2016; Liu et al. 2017; Wang et al. 2018; Deng et al. 2019) which emphasize the importance of learning features which are both separable and discriminative, through the choice of a suitable loss function. Learning discriminative features is not only important for open-set recognition (Liu et al. 2017) 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 (Wen et al. 2016). To this end, FaceNet (Schroff, Kalenichenko, and Philbin 2015) 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 (Wen et al. 2016) (called Center loss) 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 (Liu et al. 2017), 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 distances are smaller than inter-class distances. Following the idea of working in the angular domain, the more recent works of CosFace (Wang et al. 2018) and ArcFace (Deng et al. 2019) further define concepts of "center" and "margin" to obtain highly discriminative features for face recognition.

Face normalization and pose augmentation. Beyond different losses, another line of work 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 (Taigman et al. 2014). However, frontalization for the case of large poses is a difficult problem. To handle large pose variations, the work of (Masi et al. 2016) 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 (Masi et al. 2016). Using 3DMMs for conditioning GANs for large pose frontalization is proposed in (Yin et al. 2017). Simultaneously learning pose-invariant identity features and synthesizing faces in arbitrary poses is proposed in (Tran, Yin, and Liu 2017). (Deng et al. 2018) 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 FAN-Face

3.1 Overview

Our method is based on integration of features from 2 networks: a facial landmark localization network and a face recognition network. The facial landmark localization network is a pre-trained FAN (Bulat and Tzimiropoulos 2017b) 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) 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 (Newell, Yang, and Deng 2016) built using the residual block of (Bulat and Tzimiropoulos 2017a). We used 2 stacks as they suffice for good accuracy. The face recognition network, denoted as FRN, is a ResNet (He et al. 2016) 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 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 subsections. An overview of our method is illustrated in Fig. 1.

3.2 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

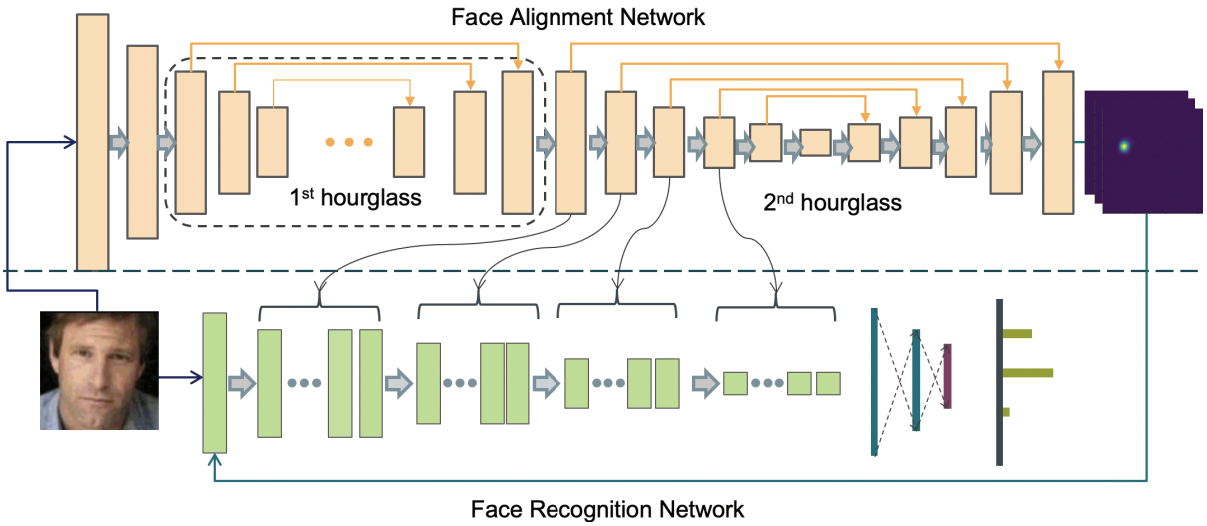

Figure 1: Overview of our method: We use a pre-trained Face Alignment Network (FAN) to extract features and landmark heatmaps from the input image. The heatmaps are firstly stacked along with the image and then passed as input to a Face Recognition Network (FRN). The features (here taken from the high-to-low part of the 2-nd hourglass of FAN) are gradually integrated with features computed by FRN. Fig. 2 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 Fig. 3.

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:

- it can be used to establish landmark correspondence across different face images.
- it captures the spatial configuration of all landmarks, and hence it captures pose, expression and shape information.
- as each heatmap is a confidence map, a number of works have shown that it also provides spatial context and part relationships (Wei et al. 2016).

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 stacked image-heatmap representation:

$$I_H \in R^{(C+N) \times M_I \times M_I}, \tag{1}$$

is used as input to train the FRN. See also Fig. 1. 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 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 (Jackson et al. 2017). However, to our knowledge, we are the first to investigate its usefulness for the *high*-level task of face recognition.

3.3 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:

$$\tilde{y}_k = y_k + \gamma f(x_l, y_k), \tag{2}$$

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 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. We always select the feature tensor at the end of each stage before downsampling. Two instantiations of this idea are as follows:

1-1 connectivity. In this integration scheme, we combine 1 feature tensor from FAN with 1 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 5) that the high-to-low part provides better features

for face recognition. Fig. 2 shows how the two networks are integrated under this scheme when FRN is a ResNet-34.

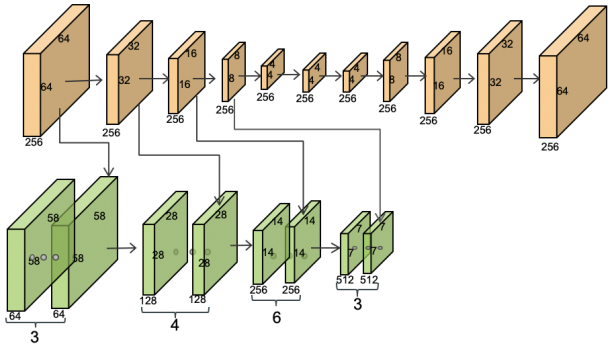

Figure 2: Proposed 1:1 connectivity between FAN and FRN: at each spatial resolution, a feature tensor from the high-to-low part of the hourglass (shown in top) is combined with a feature tensor from FRN (a ResNet-34 in this example shown in bottom). The features are combined with the integration layer of Fig. 3 and used as input to the next layer of FRN.

1-many connectivity. In this integration scheme, we combine 1 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

3.4 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

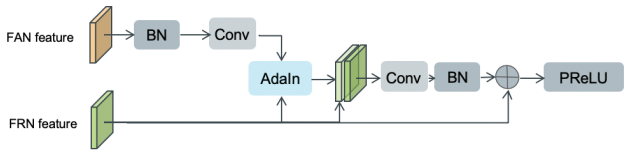

Figure 3: The proposed integration layer. FAN features are processed by a batch normalization layer that adjusts their scale followed by a 1×1 conv. layer that further aligns them with FRN features. Then, An Adaptive Instance Norm makes the distribution of the two features similar. The two features are 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.

(denoted as BL) is depicted in Fig. 3 and has the following main features: (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 y_k to be similar to that of x_l. 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.

We also propose 2 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 5, we compare all 3 layers, namely BL, SL, and AL.

3.5 Model Size and Computational Cost

It is important to note that the integration layers add only a very small number of extra parameters to FRN. For example, compared to the ResNet-34 baseline, the extra parameters added by our method, depending on the FAN-FRN connectivity and the integration layer used, are from 40K to 110K. Note that most face recognition methods including ArcFace use facial landmark detection for pre-processing. Our integration layer re-uses features from such a network by adding minimal computational cost.

4 Training and Implementation Details

Loss functions. To train our networks, we mostly used the ArcFace loss (Deng et al. 2019) which has been shown to outperform all other recently proposed methods like (Wen et al. 2016; Liu et al. 2017; Wang et al. 2018). This is important because we show systematic improvements on top of (Deng et al. 2019) 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 (Cao et al. 2018) (an improved version of VGGFace (Parkhi et al. 2015)), 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 (Whitelam et al. 2017) and IJB-C (Maze et al. 2018) datasets. Besides, we trained our model on MS1MV2 (Deng et al. 2019), a semi-automatically refined version of MS-Celeb-1M dataset (Guo et al. 2016) 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 (Wu et al. 2018). There are 79, 077 identities and 5 million images remaining. We also trained our model on CASIA-Webface (Yi et al. 2014) which contains 0.49M face images from 10,575 subjects. This model was evaluated on LFW (Huang et al. 2008), YTF (Wolf, Hassner, and Maoz 2011) and MegaFace (Kemelmacher-Shlizerman et al. 2016). In our experiments, we removed the images that belong to identities from the testing datasets.

	Method	10^{-5}	10^{-4}
Baseline	ArcFace (ResNet-34)	0.747	0.859
AS1	H2	0.750	0.864
AS1	H2+	0.771	0.866
AS2	H2,1-1,BL,l2h-2	0.752	0.871
AS2	H2,1-1,BL,h2l-2	**0.772**	0.876
AS3	H2,1-1,AL,h2l-2	0.766	0.873
AS3	H2,1-1,SL,h2l-2	0.739	0.857
AS4	H2,1M,BL,h2l-2	0.761	0.874
AS5	ResNet-34-Softmax	0.619	0.786
AS5	H2,1-1,BL,h2l-2-Softmax	0.657	0.796
AS6	H2, 1-1, BL, Wing	0.769	0.870
AS6	H2, 1-1, BL, Simple	0.770	0.874
AS6	H2, 1-1, BL, HRNet	0.771	**0.877**
AS7	Multi-task(2018)	0.583	0.731

Table 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 5.

Other hyperparameters. We followed the publicly available code of (Deng et al. 2019) for implementing and training our models. For a fair comparison, we used the same ResNet as ArcFace. 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 (Paszke et al. 2017).

Face pre-processing. We followed standard practices in face recognition (Wen et al. 2016; Liu et al. 2017; Wang et al. 2018; Deng et al. 2019) 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.

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 ResNet-34, and unless otherwise stated all methods are trained with the ArcFace loss. All results are shown in Table 1. We provide results for a wide range of False Acceptance Rates (FARs), however, when we compare methods in the sections below, we primarily base our evaluations on True Acceptance Rate (TAR) at FAR=10^{-4} as in (Deng et al. 2019). 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 slighly worse result). We call FAN-Face(H2), the variant of our model, obtained by simple stacking these heatmaps with FRN (see Section 3.2). As Table 1 shows, this gives some decent improvement over the baseline ArcFace. We also investigated whether stacking heatmaps with features from various layers of FRN is also beneficial. To this end, we stacked the heatmaps from hourglass 2 (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 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}.

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 3.3) and the Basic Layer (see Section 3.4). We call this variant FAN-Face(H2, 1-1, BL). We wanted to quantify which of the h2l or l2h subnetworks is superior. To this end, we chose the h2l and l2h 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 1 shows the results: the h2l subnetwork clearly provides much better features for integration than those of l2h. This is expected as l2h features resemble heatmaps, and we are already using heatmap information directly.

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 3.4). The results are shown in Table 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, 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 3.3). The results are shown in Table 1. We observe that 1-M offers no improvement over 1-1.

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 1. We observe that with ArcFace loss, our method improves upon the baseline even more. This emphasizes the importance of having a good loss, and the complementary 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 ResNet-34 trained in-house to regress x,y coordinates as in (Feng et al. 2018), a ResNet-34 trained in-house to regress heatmaps as in (Xiao, Wu, and Wei 2018) and the pre-trained SOTA network of (Sun et al. 2019). Results vary a bit because these networks are not equally accurate but, 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 have an encoder similar to hourglass, so they can all be integrated within our framework.

For example, for H2,1-1,BL, Simple where (Xiao, Wu, and Wei 2018) is used to replace FAN, integration is straightforward as the encoder is a ResNet-34 so it has the same

structure as our FRN. Similar to FAN, we do feature integration by taking the features at the end of each stage/module before downsampling. We also do heatmap integration using the predicted heatmaps. It is worth mentioning that (Feng et al. 2018), also based on ResNet-34, regresses x,y coords but heatmaps can be readily regenerated from them.

AS7: Multi-task (Yin and Liu 2018). A natural comparison that comes to mind is between our method and an FRN that has a second head to also predict the landmarks in a multi-task fashion, see for example (Yin and Liu 2018). For the sake of a fair comparison, we preserved the ResNet-34 structure for this method and added a heatmap prediction head after layer 4 (of ResNet-34). 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 1 shows the obtained results. We observe that multi-task learning does not offer competitive performance.

Visualizations. Some examples of feature maps extracted in early layers produced by our model and by ArcFace are shown in Fig. 4. While some feature maps are similar there are also a few that focus on facial landmarks.

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.

Our models used: We provide the results provided by our best model FAN-Face(H2, 1-1, BL, h2l-2) which uses heatmaps from hourglass 2, integrates features with 1-1 connectivity from the h2l subnetwork of hourglass 2, and uses as integration layer the Basic Layer of Section 3.4. We provide results using both ResNet-34 and ResNet-50 for FRN.

Our in-house baselines: As a very strong baseline, for all experiments, we used our implementation of ArcFace, using both ResNet-34 and ResNet-50. Our implementation (based on the code provided in (Deng et al. 2019)) gives slightly better results than the ones reported in the original paper. Besides, our implementation of CosFace (Wang et al. 2018) with ResNet-50 was also compared because it was a strong baseline in (Deng et al. 2019).

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. ResNet-50, ResNet-34, VGG) and the training set used.

The IJB-B dataset (Whitelam et al. 2017) 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 (Maze et al. 2018) 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 2 and 3, we can observe that, for both datasets,, our methods provide consistently the best performance at *FAR*=10^{-5}, outperforming both previously proposed methods and in-house baselines, by a large margin. Notably, our ResNet-34 model provides performance which is better or in par with previously state-of-the-art ResNet-50-based models.

Method	10^{-5}	10^{-4}	10^{-3}	10^{-2}
R50 (2018)	0.647	0.784	0.878	0.938
SE50 (2018)	0.671	0.800	0.888	0.949
R50+SE50 (2018)	-	0.800	0.887	0.946
MNv (R50) (2018)	0.683	0.818	0.902	0.955
MNvc (R50) (2018)	0.708	0.831	0.909	0.958
R50+DCN(Kpts) (2018)	-	0.850	0.927	0.970
R50+DCN(Divs) (2018)	-	0.841	0.930	0.972
SE50+DCN(Kpts) (2018)	-	0.846	0.935	0.974
SE50+DCN(Divs) (2018)	-	0.849	0.937	0.975
ArcFace (R50) (2019)	0.812	0.898	0.944	**0.976**
ArcFace (R34, in-house)	0.775	0.885	0.940	0.973
Ours (R34)	0.817	0.900	0.945	0.974
CosFace (R50, in-house)	0.806	0.895	0.945	0.972
ArcFace (R50, in-house)	0.814	0.902	0.946	0.974
Ours (R50)	**0.835**	**0.911**	**0.947**	0.975

Table 2: Evaluation of different methods for 1:1 verification on IJB-B dataset. R34 and R50 denote ResNet-34 and ResNet-50. SE50 denotes SENet-50. All methods were trained on VGGFace2 dataset.

Method	10^{-5}	10^{-4}	10^{-3}	10^{-2}
R50 (2018)	0.734	0.825	0.900	0.950
SE50 (2018)	0.747	0.840	0.910	0.960
R50+SE50 (2018)	-	0.841	0.909	0.957
MNv (R50) (2018)	0.755	0.852	0.920	0.965
MNvc (R50) (2018)	0.771	0.862	0.927	0.968
R50+DCN(Kpts) (2018)	-	0.867	0.940	0.979
R50+DCN(Divs) (2018)	-	0.880	0.944	0.981
SE50+DCN(Kpts) (2018)	-	0.874	0.944	0.981
SE50+DCN(Divs) (2018)	-	0.885	0.947	**0.983**
ArcFace (2019)	0.861	0.921	0.959	0.982
ArcFace (R34, in-house)	0.851	0.912	0.955	0.981
Ours(R34)	0.873	0.926	**0.960**	0.981
CosFace (R50, in-house)	0.864	0.918	0.952	0.980
ArcFace (R50, in-house)	0.872	0.924	0.959	0.982
Ours(R50)	**0.874**	**0.935**	0.959	0.982

Table 3: Evaluation of different methods for 1:1 verification on IJB-C dataset. R34 and R50 denote ResNet-34 and ResNet-50. SE50 denotes SENet-50. All methods were trained on VGGFace2 dataset.

We also conducted the MS1MV2 experiment reported in (Deng et al. 2019) using ResNet-100. Please see Table 4.

Ms/Ds	IJB-B			IJB-C		
	Ver	Id		Ver	Id	
	10^{-4}	rk=1	rk=5	10^{-4}	rk=1	rk=5
ArcFace	94.5	93.2	95.8	95.9	94.4	96.5
Ours	**95.4**	**93.7**	**96.1**	**96.8**	**94.8**	**97.0**

Table 4: 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. rk-1 is rank-1 and rk-5 is rank-5.

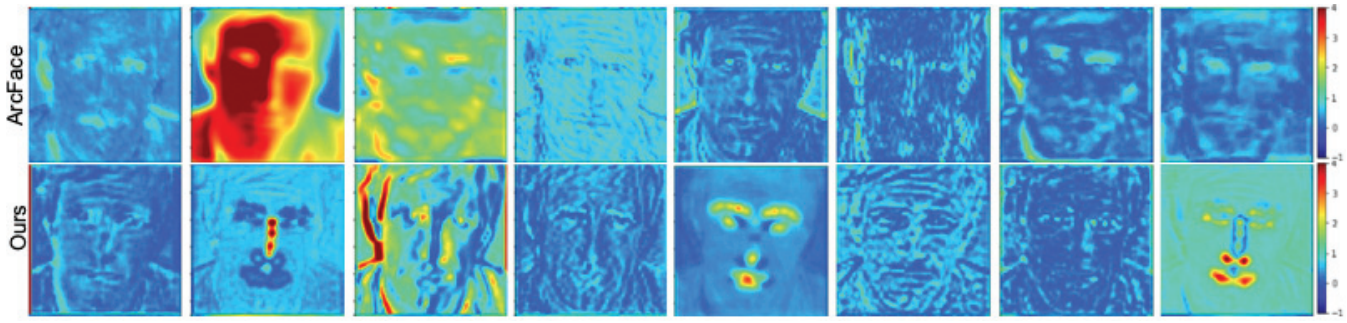

Figure 4: Visualization of feature maps from ArcFace (shown in top) and our model (shown in bottom). By using FAN features to guide FRN learning, facial landmark related attention is added to the learned features.

Method	Network	Id (%)	Ver (%)
Softmax (2017)	ResNet-64	54.86	65.93
SphereFace (2017)	ResNet-64	72.73	85.56
CosFace (2018)	ResNet-64	77.11	89.88
ArcFace (2019)	ResNet-50	77.50	92.34
ArcFace (in-house)	ResNet-34	75.52	89.53
Ours	ResNet-34	77.54	92.06
CosFace (in-house)	ResNet-50	75.93	91.02
ArcFace (in-house)	ResNet-50	76.44	91.44
Ours	ResNet-50	**78.32**	**92.83**

Table 5: Identification and verification results on MegaFace Challenge 1. Id refers to rank-1 face identification accuracy and Ver refers to face verification TAR (True Acceptance Rate) at 10^{-6} FAR (False Acceptance Rate). All methods were trained on CASIA dataset.

Method	Network	LFW	YTF
Softmax (2017)	ResNet-64	97.88	93.1
HiReST-9 (2017)	AlexNet (2012)	99.03	95.4
SphereFace (2017)	ResNet-64	99.42	95.0
CosFace (2018)	ResNet-64	99.33	96.1
ArcFace (2019)	ResNet-50	99.53	-
ArcFace (in-house)	ResNet-34	99.40	95.42
Ours	ResNet-34	99.52	96.34
CosFace (in-house)	ResNet-50	99.30	95.78
ArcFace (in-house)	ResNet-50	99.47	96.13
Ours	ResNet-50	**99.56**	**96.72**

Table 6: Verification performance (%) on LFW and YTF datasets.

6.1 MegaFace Dataset

MegaFace (Kemelmacher-Shlizerman et al. 2016) 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 (Ng and Winkler 2014) 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.

Table 5 shows the obtained results. We observe that our ResNet-50 model outperforms all previous methods and in-house baselines, significantly. Again, our ResNet-34 model is the second best performing model, slightly outperforming the original implementation of ArcFace (Deng et al. 2019) for identification which however used a ResNet-50 model.

6.2 LFW and YTF Datasets

We also report results on LFW (Huang et al. 2008) (13,233 web-collected images from 5,749 individuals) and YTF (Wolf, Hassner, and Maoz 2011) (3,425 videos from 1,595 different identities). As typical in literature, our models and in-house baselines, were trained on CASIA dataset (Yi et al. 2014). Following the unrestricted protocol with labelled outside data (Huang and Learned-Miller 2014), we report the performance our models on 6,000 face pairs from LFW and 5,000 videos pairs from YTF in Table 6. As in our previous experiments, our ResNet-50 model performs the best while our ResNet-34 is the second best along with the ArcFace ResNet-50-based model of (Deng et al. 2019).

7 Conclusions

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 localization network are integrated into the face recognition feature extraction process to provide face related information and establish correspondence for face matching. We 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 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.

References

Bulat, A., and Tzimiropoulos, G. 2017a. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *ICCV*.

Bulat, A., and Tzimiropoulos, G. 2017b. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*.

Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. Vggface2: A dataset for recognising faces across pose and age. In *FG*.

Deng, J.; Cheng, S.; Xue, N.; Zhou, Y.; and Zafeiriou, S. 2018. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.

Feng, Z.-H.; Kittler, J.; Awais, M.; Huber, P.; and Wu, X.-J. 2018. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR*.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, G. B., and Learned-Miller, E. 2014. Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*.

Huang, G. B.; Mattar, M.; Berg, T.; and Learned-Miller, E. 2008. 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*.

Jackson, A. S.; Bulat, A.; Argyriou, V.; and Tzimiropoulos, G. 2017. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *ICCV*.

Kemelmacher-Shlizerman, I.; Seitz, S. M.; Miller, D.; and Brossard, E. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*.

Masi, I.; Rawls, S.; Medioni, G.; and Natarajan, P. 2016. Pose-aware face recognition in the wild. In *CVPR*.

Maze, B.; Adams, J.; Duncan, J. A.; Kalka, N.; Miller, T.; Otto, C.; Jain, A. K.; Niggel, W. T.; Anderson, J.; Cheney, J.; et al. 2018. Iarpa janus benchmark–c: Face dataset and protocol. In *ICB*.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *ECCV*.

Ng, H.-W., and Winkler, S. 2014. A data-driven approach to cleaning large face datasets. In *ICIP*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; et al. 2015. Deep face recognition. In *BMVC*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *CVPR*.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.

Tran, L.; Yin, X.; and Liu, X. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*.

Wei, S.-E.; Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2016. Convolutional pose machines. In *CVPR*.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.

Whitelam, C.; Taborsky, E.; Blanton, A.; Maze, B.; Adams, J.; Miller, T.; Kalka, N.; Jain, A. K.; Duncan, J. A.; Allen, K.; et al. 2017. Iarpa janus benchmark-b face dataset. In *CVPRW*.

Wolf, L.; Hassner, T.; and Maoz, I. 2011. Face recognition in unconstrained videos with matched background similarity. In *CVPR*.

Wu, W.; Kan, M.; Liu, X.; Yang, Y.; Shan, S.; and Chen, X. 2017. Recursive spatial transformer (rest) for alignment-free face recognition. In *ICCV*.

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

Xiao, B.; Wu, H.; and Wei, Y. 2018. Simple baselines for human pose estimation and tracking. In *ECCV*.

Xie, W., and Zisserman, A. 2018. Multicolumn networks for face recognition. *BMVC*.

Xie, W.; Shen, L.; and Zisserman, A. 2018. Comparator networks. In *ECCV*.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. *arXiv*.

Yin, X., and Liu, X. 2018. Multi-task convolutional neural network for pose-invariant face recognition. *TIP*.

Yin, X.; Yu, X.; Sohn, K.; Liu, X.; and Chandraker, M. 2017. Towards large-pose face frontalization in the wild. In *ICCV*.