Searching for Alignment in Face Recognition

Xiaqing Xu[1], Qiang Meng[1], Yunxiao Qin[2], Jianzhu Guo[3,4], Chenxu Zhao[5*], Feng Zhou[1], Zhen Lei[3,4]

[1]AIBEE, Beijing, China, [2]Northwestern Polytechnical University, Xian, China
[3] CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[4] School of Artificial Intelligence, University of Chinese Academy of Sciences
[5]Academy of Sciences, Mininglamp Technology, Beijing, China
{xqxu; qmeng; fzhou}@aibee.com, qyxqyx@mail.nwpu.edu.cn, {jianzhu.guo; zlei}@nlpr.ia.ac.cn
zhaochenxu@mininglamp.com

Abstract

A standard pipeline of current face recognition frameworks consists of four individual steps: locating a face with a rough bounding box and several fiducial landmarks, aligning the face image using a pre-defined template, extracting representations and comparing. Among them, face detection, landmark detection and representation learning have long been studied and a lot of works have been proposed. As an essential step with a significant impact on recognition performance, the alignment step has attracted little attention. In this paper, we first explore and highlight the effects of different alignment templates on face recognition. Then, for the first time, we try to search for the optimal template automatically. We construct a well-defined searching space by decomposing the template searching into the *crop size* and *vertical shift*, and propose an efficient method **F**ace **A**lignment **P**olicy **S**earch (FAPS). Besides, a well-designed benchmark is proposed to evaluate the searched policy. Experiments on our proposed benchmark validate the effectiveness of our method to improve face recognition performance.

Introduction

Face recognition is a long-standing topic in the research community of computer vision. A standard pipeline of the recognition framework consists of four individual steps: locating faces with bounding boxes and fiducial points, aligning face images using a pre-defined template, extracting face representations and representation comparing. The second step, also named as face alignment (in Fig. 2), serves as deforming face images such that fiducial points are spatially aligned and simplifies the recognition task by normalizing the in-plane rotation, scale and translation variations. However, most recent works (Taigman et al. 2014a; Sun, Wang, and Tang 2014a; Schroff, Kalenichenko, and Philbin 2015; Liu et al. 2017; Deng et al. 2019; Kang et al. 2019) on face recognition focus on designing loss functions and exploring network structures. In contrast, the alignment procedure before model training is less studied.

In this paper, we first explore the effects of the alignment templates(Deng et al. 2019; Zhu et al. 2019; Guo et al. 2020b) on face recognition performance. Face features can be divided into two sets depending on the zone where they are located: internal features, including eyes, nose and mouth, and external features, composed by the hair, chin and face outline. The benefits of external information have been observed in some early works (Lapedriza, Masip, and Vitria 2005; Andrews et al. 2010), but they are rarely discussed in the modern face recognition framework (Taigman et al. 2014a; Schroff, Kalenichenko, and Philbin 2015; Liu et al. 2017; Deng et al. 2019). Significant differences in the 1v1 results are observed by using templates with different degrees of external features involved, as illustrated in Fig. 1. An open problem arises: *is there an optimal template such that the produced face region gives the best recognition performance?* Specifically, it remains unknown whether fewer backgrounds or irrelevant textures to face (e.g., hair, forehead) benefit face recognition. Besides, it is unclear whether the optimal template generalizes well across various conditions including the pose, age and illumination.

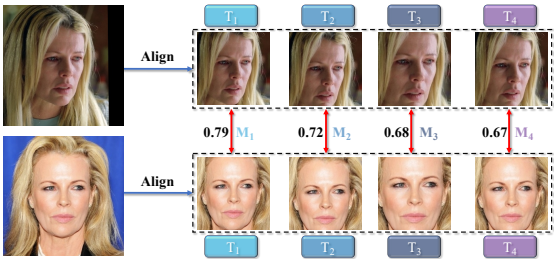

Figure 1: Verification results based on different face templates. Models $M_i, i = 1, 2, 3, 4$ are trained with samples aligned by templates $T_i, i = 1, 2, 3, 4$ respectively. Significant differences between cosine similarities are observed.

Instead of manually designing templates, we propose to automate the process of finding the optimal template for recognition. To this end, we decompose differences of templates into *vertical shift* and *crop size*, and construct a well-defined discrete searching space. We call the *vertical shift* and *crop size* pair an alignment policy. The equivalence relation of the alignment policy and the template is described and proved in Section Face Alignment, and illustrated in Fig. 2. The template searching space is thus projected to the cropping box space spanned by *vertical shift* and *crop size*.

*Corresponding Author.

A straightforward way to search for the template is using the grid search. However, grid search is inefficient and costly. For example, the total size of searching space in our work is 93 and the grid search for the optimal template on the dataset like CASIA (Yi et al. 2014) is rather time-consuming (costs about **9102** GPU hours with 8 Tesla V100 GPUs).

In this paper, we propose an evolution-based method named Face Alignment Policy Search (FAPS) to efficiently searches for the optimal template. FAPS jointly trains a population of models with evolving templates. Inspired by PBT (Jaderberg et al. 2017), we reuse the partially trained weights to accelerate the searching procedure, as training from scratch on a large-scale dataset is time-consuming. To improve the generality of the partially trained model, we set the upper bound of search space as *SuperROI* such that the models have the knowledge of all the facial parts and can concentrate on the more informational area. The original *explore* in PBT mainly considers perturbing the hyperparameter from a better-performing population or resampling new hyperparameter from originally defined distribution, while ignores the relations among different templates in our problem. To accelerate the discovering of better *crop size* and *vertical shift*, we propose *Intersection based Crossover* to combine the strength of well-performing templates (Fig. 5).

Until now, searching for alignment in face recognition is less-studied and there exists no common protocol for evaluation, thus we introduce a well-designed benchmark(including LFW (Huang et al. 2008), AgeDB-30 (Moschoglou et al. 2017) and MultiPIE (Gross et al. 2010), *etc.*) to evaluate the searched face crop template.

Our main contributions include: (i) To the best of our knowledge, we explore and highlight the effects of alignment templates on face recognition for the first time. (ii) We construct a well-defined searching space by decomposing the template searching into *crop size* and *vertical shift* searching, and propose an efficient method named FAPS for template searching. (iii) A well-designed benchmark is proposed to evaluate the searched policy. Extensive experiments on the proposed benchmark validate the efficacy of FAPS.

Background

Face Alignment

Face alignment is used to align faces to a unified distribution and reduce the geometric variations. The most commonly adopted way is applying a 2D affine transformation to calibrate facial landmarks to predefined 2D (Wang et al. 2018; Deng et al. 2019; Wang et al. 2017; Liu et al. 2017) or 3D templates (Taigman et al. 2014b; Guo et al. 2020a).

Besides the affine transformation, some other works learn non-rigid transformations. For example, ReST (Wu et al. 2017) introduces a recursive spatial transformer to learn complex transformation. (Zhou, Cao, and Sun 2018) use local homography transformations estimated by a rectification network to rectify faces. These methods aim for alignment-free through learning alignment jointly with the recognition network in an end-to-end fashion. Despite their achievements, additional computational cost and loss of identity information limit their usage in real-world applications.

Apart from the types of transformation, another critical element of alignment is how to design a proper facial template. A suitable template should focus on facial features that benefit the recognition most. Some early works (Lapedriza, Masip, and Vitria 2005; Andrews et al. 2010) have observed performance improvements when including some external face features (*i.e.*, hair, chin and face outline) compared to using internal face features alone (*i.e.*, eyes, nose and mouth). One optimal solution is to apply multi-patches methods (Sun, Wang, and Tang 2014b; Sun et al. 2014; Sun, Wang, and Tang 2014c; Liu et al. 2015) which process an image via multiple templates and dump them to different recognition models. Although this strategy improves performances, it requires too much additional computational costs and carefully designed ensemble methods. In our work, we compare the performance of a set of templates and aim to find the optimal one for the face recognition task.

Hyperparameter Optimization

As face alignment policy is a hyperparameter for face recognition, our work closely correlates with the hyperparameter optimization(Cubuk et al. 2019; Lim et al. 2019; Ryuichiro et al. 2019; Zhang et al. 2020) problem which automatically tunes the hyperparameters. An RL-based method called AutoAugment (Cubuk et al. 2019) is proposed to train a controller to search for the best data augmentation policy based on specific datasets and models. Apart from the RL-based methods, evolution-based methods (Jaderberg et al. 2017; Ho et al. 2019) spring recently. For example, PBT (Jaderberg et al. 2017) jointly trains a population of models and searches for their hyperparameters with evolution to improve the models' performances. *Exploit* and *explore* are the two most important strategies of PBT. *Exploit* is responsible for copying better weights and hyperparameters from a well-performing model to the inferior one. *Explore* creates new hyperparameters for the poor-performing model by either resampling new hyperparameters from the originally defined prior distribution or perturbing the copied hyperparameters from a well-performing model. These two strategies make PBT faster and more effective.

In this work, inspired by PBT, we develop a novel evolution-based method named FAPS to search for a better face alignment strategy. The *exploit* and *explore* from PBT are also adopted in our method.

Methodology

In this section, we first review the face alignment process via 2D affine transformations and demonstrate that template searching can be decomposed into searching *crop size* and *vertical shift*. Then we detail the proposed FAPS.

Face Alignment

We define one alignment template as a composition of landmarks R_i with cropped area $[0, 0, w_b, w_b]$ (a $w_b \times w_b$ rectangle with top left point [0, 0]). In this work, facial landmarks in all templates share the same shape. To be more specific, any R_i can be transformed from one base landmarks R_0 by scaling s_i and shifting x_i, y_i over the x, y axis respectively as shown in Fig. 2.

One face image $\mathbf{I}$ is aligned to landmarks R_i by a 2D affine transformation $\mathbf{T}$. Denote $\mathbf{I}_i^a$ as the transferred image based on landmarks R_i. We seek an optimized affine

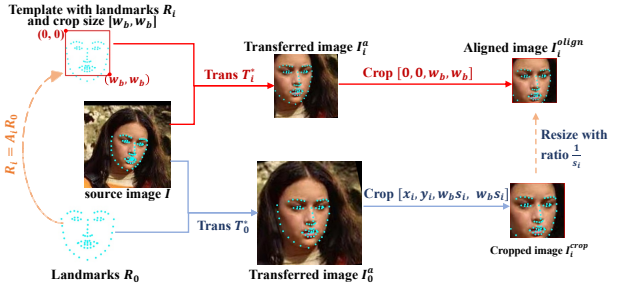

Figure 2: An overview of the face alignment process. Assuming we have a template with landmarks R_i and cropping rectangle from point (0, 0) to point (w_b, w_b). R_i can be transferred from R_o by scaling s_i and shifting $[x_i, y_i]$, *i.e.,* $R_i = \mathbf{A}_i R_0 = \begin{bmatrix} s_i & -s_i & x_i \\ s_i & s_i & y_i \\ 0 & 0 & 1 \end{bmatrix} R_0$. The source image $\mathbf{I}$ is transferred to $\mathbf{I}_0^a$ and $\mathbf{I}_i^a$ based on landmarks R_0 and R_i respectively. We prove that the result $\mathbf{I}_i^{align}$ aligned by the current template is the same as resizing cropped image $\mathbf{I}_0^{crop}$. Therefore, the aligned image from an arbitrary template can be got by cropping and resizing from the same image $\mathbf{I}_0^a$.

transformation matrix $\mathbf{T}_i^*$ to transfer a face image $\mathbf{I}$ to $\mathbf{I}_i^a$. It can be proved that $\mathbf{T}_i^* = \mathbf{A}_i \mathbf{T}_0^*$. Then we have $\mathbf{I}_i^a = \mathbf{T}^*[\mathbf{I}] = \mathbf{A}_i \mathbf{T}_0^*[\mathbf{I}] = \mathbf{A}_i \mathbf{I}_0^a$, which shows that the transferred image based on landmarks R_i can be achieved by performing transformation $\mathbf{A}_i$ on the $\mathbf{I}_0^a$. The final aligned image is the area $[0, 0, w_b, w_b]$ of transferred image $\mathbf{I}_i^a$, which is given by the following steps: 1) Transfer image $\mathbf{I}$ to $\mathbf{I}_0^a$ based on the base landmarks R_0. 2) Crop the image with area $[x_i, y_i, w_b \cdot s_i, w_b \cdot s_i]$. 3) Resize the area by size $[w_b, w_b]$.

Therefore, instead of designing various templates and aligning a face multiple times, we simplify the processes by aligning once by the base template R_0 and operating (crop + resize) on the same image $\mathbf{I}_0^a$. In our implementation, landmarks in all templates are placed to be horizontally symmetric, which makes $x_i = 0$. Let $m_i = w_b \cdot s_i$, $\delta_i = y_i / s_i$, our target now is to find the optimal m^*, δ^*. We call $\boldsymbol{p} = \{m, \delta\}$ an alignment policy and each policy represents a corresponding template.

Search Space

To facilitate the search process, we place the base face landmarks R_0 to a 300×300 canvas with the mid-point of the nose (red point in Figure 3(a)) at the center. We denote this template as T_p. After aligning an image to R_0, FAPS searches for the optimal region to simulate the effects of applying different templates. A candidate region is determined by 1) *crop size* m which controls the tightness of cropped face and 2) *vertical shift* δ which controls the center of cropped area. Some examples are presented in Fig. 3(c).

Denote $\mathcal{P}$ as the union of all candidate $\boldsymbol{p}$, *i.e.,* the search space. We define the search space as follows: With upper bound m_{max} and $\delta = 0$, the selected region is able to cover both internal and external face features (Fig. 3(b)). While with m_{min} and $\delta = 0$, only indispensable facial parts (eyes, nose, mouth) are kept as shown in Fig. 3(c).

Through the variation of vertical shift δ, some facial features are dropped and some new features are included in the input. When m is set to the smallest scale m_{min},

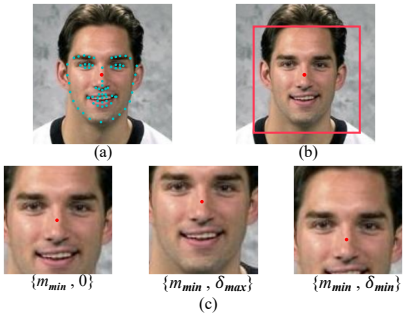

Figure 3: An overview of the search space: (a) The face image is 300×300 after aligned with base landmarks R_0. The red landmark point is placed in the center of the canvas. (b) The red box ($\{m = m_{max}, \delta = 0\}$) shows the upper bound of the search space. (c) The input facial image varies rapidly with different δ and fixed crop size m_{min}. When δ is 0, indispensable facial parts (eyes, nose, mouth) and half of the forehead and chin are kept. The forehead is almost removed when $\delta = \delta_{max}$. When setting δ to δ_{min}, the forehead is well-preserved while the chin is dropped.

this phenomenon becomes more obvious (Fig. 3(c)). If δ is set to the maximum value δ_{max}, only the eyebrows are preserved, the forehead is almost omitted. When δ is set to the minimum value δ_{min}, only the mouth is preserved, the chin is dropped. With such an extreme setting of δ, the importance of different facial areas can be discovered.

Search Strategy

Denote the recognition model as f and its weights as w, we represent model trained with images aligned by $\boldsymbol{p}$ as $f(w|\boldsymbol{p})$. Let $\mathcal{L}_{train}$ and ACC_{val} be the training loss and validation accuracy, respectively. The process of finding the optimal alignment policy can be formulated as:

$$\boldsymbol{p}^* = \text{argmax}_{\boldsymbol{p} \in \mathcal{P}} \text{ACC}_{val}(f(w^*|\boldsymbol{p})) \quad (1)$$

$$s.t.\ w^* = \text{argmin}_w\ \mathcal{L}_{train}\ f(w|\boldsymbol{p}) \quad (2)$$

To find the optimal solution, the trivial approach like grid search is to traverse all possible $\boldsymbol{p}$. In this way, model f needs to be trained $|\mathcal{P}|$ times, which is time-consuming and inefficient. Inspired by Population based Training (PBT)(Jaderberg et al. 2017), we train a fixed population of models with different $\boldsymbol{p}$ in parallel. The "exploit-and-explore" procedure is applied to the worse performing models at a certain interval, where the inferior model clones the weight of better performing model and updates the alignment policy through perturbing this well-performing model's $\boldsymbol{p}$. The model can be trained with a new $\boldsymbol{p}$ without reinitialized. The total computation is largely reduced to a single optimization process (Fig. 4).

SuperROI To improve the generality of partial trained model when cloning the weights, we initialize $\boldsymbol{p}$ to $\{m_{max}, 0\}$ as shown in Fig. 3 (b), *i.e.*, an initialized Region of Interest (ROI) containing all internal features (eyes, nose and mouth) and external features (jaw-line, ears, part of the hair, *etc.*). Under this setting, beginning models can have the capacity to handle information from all facial parts. When switching to other policies, the facial region can be a part of the initial one and no new facial parts are introduced. Models only need to learn the trade-offs from current features, *i.e.*,

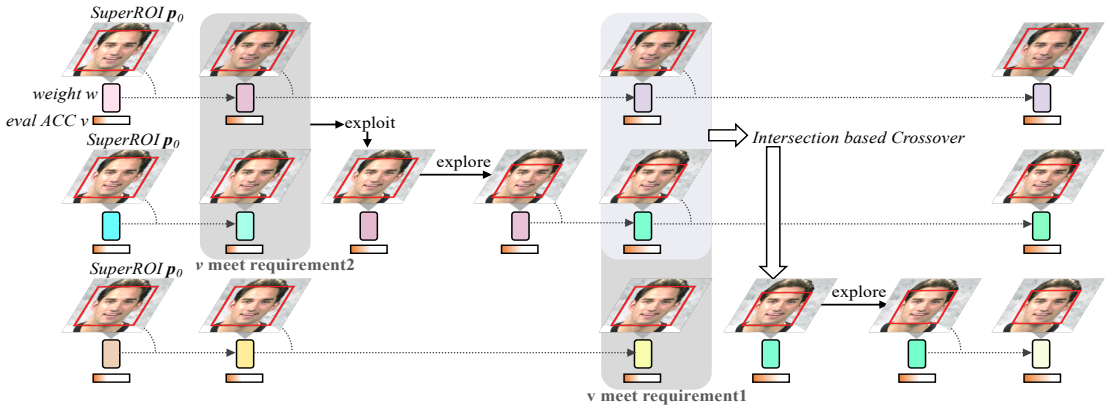

Figure 4: Overview of the proposed FAPS. We first initialize a fixed population of models with *SuperROI* $\boldsymbol{p}_0$. After each epoch, each model's accuracy v on the validation set is calculated. If an under-performing model meets *requirement1*, the *Intersection based Crossover* will be operated on the model. Then a new alignment policy is generated by combining the policies of two well-performing models. If an inferior model meets *requirement2*, *exploit* and *explore* will be performed. To be more specific, model weights are copied by those of a superior model and new alignment policy is generated by disturbing a superior policy.

learn to focus on remaining facial parts and ignore removed ones. This process shares the spirit of the supernet in Neural Architecture Search (Chen et al. 2019; Guo et al. 2019; Chu et al. 2019), consequently, we name $\boldsymbol{p}_0 = \{m_{max}, 0\}$ as *SuperROI*.

Intersection based Crossover The original *explore* of PBT either re-samples new hyperparameter directly from the originally defined prior distribution or perturbs the current hyperparameter from a well-behaved population to upgrade the weak-behaved population. The former strategy, which resembles random search (Bergstra and Bengio 2012), can relieve the problem of local minima but cannot guarantee qualities of sampled hyperparameters. The later strategy is analogous to the mutation in genetic algorithms and has a high probability of finding better hyperparameter. However, it generates new hyperparameter depending on one particular hyperparameter each time instead of hyperparameters of well-behaved populations, which may lead to unstable results or even a local minimum. Besides the above hyperparameter generation methods, the common trend of well-behaved ones is not fully utilized.

Inspired by crossover in genetic algorithms (Spears 1993), we propose *Intersection based Crossover* to facilitate the discovering of better alignment policy $\boldsymbol{p}$ during search (Fig. 5). Suppose there exist two well-performing policies $\boldsymbol{p}_1 = \{m_1, \delta_1\}, \boldsymbol{p}_2 = \{m_2, \delta_2\}$ and the corresponding facial areas are A_1, A_2 respectively. Their intersection area $A_{1,2} = A_1 \cap A_2$ is highly possible to contain rich facial information that benefits face recognition. Policies generated by trivial crossover ($\{m_1, \delta_2\}$ and $\{m_2, \delta_1\}$) can possibly represent regions that differ a lot from both A_1, A_2, which therefore fail to cover the intersection area. Instead, *Intersection based Crossover* finds the policy whose region has the largest similarity with $A_{1,2}$. Denote $A(\boldsymbol{p})$ as the face region represented by policy $\boldsymbol{p}$ and $\textbf{iou}(A(\boldsymbol{p}), A_{1,2}) = \frac{A(\boldsymbol{p}) \cap A_{1,2}}{A(\boldsymbol{p}) \cup A_{1,2}}$, we update the policy $\boldsymbol{p}$ and model weights w by Eq.3 and Eq.4:

$$\boldsymbol{p}' \leftarrow \text{argmax}_{\boldsymbol{p} \in \mathcal{P}} \textbf{iou}(A(\boldsymbol{p}), A_{1,2}) \tag{3}$$

$$w' \leftarrow w_{i^*},\ s.t.\ i^* = \text{argmax}_{i \in \{1,2\}}\ \textbf{iou}(A(\boldsymbol{p}'), A_i) \tag{4}$$

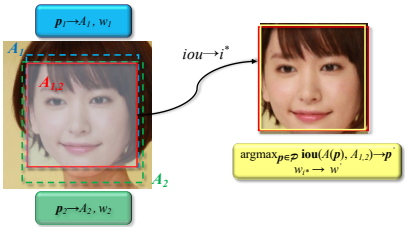

Figure 5: Illustration of *Intersection based Crossover*. $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are alignment policies of two well-performing populations. Their corresponding regions are A_1 and A_2, $A_{1,2} = A_1 \cap A_2$ represents the shared area ((red rectangle)). Our *Intersection based Crossover* finds a policy p' which has the largest IOU scores with $A_{1,2}$ (yellow rectangle). As a result, p' inherits the intersection area. The **iou** function decides whose weight can be cloned to the inferior model. The IOU score of A_1 and A' is larger, hence w_1 is chosen.

Implementation

The alignment template search process is elaborated in Algorithm 1. The details of the main function are below:

Step: In each step, we train the model in one epoch through SGD with ArcFace loss (Deng et al. 2019).

Eval: We evaluate the current model on our validation set, the verification rate is calculated as the validation accuracy.

Ready: A model is ready to go through the exploit-and-explore or *Intersection based Crossover* process once 1 epoch has elapsed.

Requirement1: The model's validation accuracy v is between the bottom $1/4$ and $3/8$ of the population.

Requirement2: The model's validation accuracy v is in the bottom $1/4$ of the population.

Exploit: Get the weight w and alignment policy $\boldsymbol{p}$ of a model that has validation accuracy v in the top $1/4$.

Explore: See Algorithm 2 for the *explore* function. For m and δ, we either perturb the original value or uniformly resample them from all possible values.

Intersection based Crossover: We choose two well-performing models $f(w_1|\boldsymbol{p}_1)$ and $f(w_2|\boldsymbol{p}_2)$ whose validation accuracies are in the top $1/4$ to generate the new alignment policy $\boldsymbol{p}'$. If $\boldsymbol{p}'$ is already deployed by the current mod-

els, an extra *explore* will be applied to $\boldsymbol{p}'$.

Algorithm 1 Face Alignment Policy Search(FAPS).

Require: Current policy search space $\mathcal{P}$, *SuperROI* $\boldsymbol{p}_0 = \{m_{max}, 0\}$, population size of models *N*.

Initialize *N* models $f(w|\boldsymbol{p}_0)$
for each model $f(w|\boldsymbol{p}_0)$ (asynchronously in parallel)
  while not end of training
    $w \leftarrow \text{step}(w|\boldsymbol{p})$ ▷ train current model with policy $\boldsymbol{p}$
    $v \leftarrow ACC_{val}(f(w|\boldsymbol{p}))$ ▷ evaluation
    if ready(f, v) **then**
      check v's performance among all models
      if v meets *requirement1* **then**
        generate w',$\boldsymbol{p}'$ via *Intersection based Crossover*
        If $\boldsymbol{p}'$ doesn't exist currently **then**
          $w, \boldsymbol{p} \leftarrow w', \boldsymbol{p}'$
        else
          $w, \boldsymbol{p} \leftarrow explore(w', \boldsymbol{p}')$
      elif v meets *requirement2* **then**
        get w',$\boldsymbol{p}'$ through *exploit*
        $w, \boldsymbol{p} \leftarrow explore(w', \boldsymbol{p}')$
      update model populations with new $f(w|\boldsymbol{p})$
return $\boldsymbol{p}$ with highest v among training

Algorithm 2 The FAPS explore function. When revising the alignment policy based on the current one, the change value is amplified by magnitude parameters.

Require: current alignment policy $\boldsymbol{p} = \{m, \delta\}$, *SuperROI*, magnitude parameters $\boldsymbol{s} = \{s_m, s_\delta\}$

for *param* in $\boldsymbol{p}$
  if random(0, 1) <0.2 **then**
    random sample *param* uniformly from search space
  else
    level = [0,1,2,3] with probability [0,1, 0.3, 0.3, 0.3]
    if random(0,1) <0.5 **then**
      $param = param - level \times \boldsymbol{s}_{param}$
    else
      $param = param + level \times \boldsymbol{s}_{param}$
    Clip *param* to stay within *SuperROI*

Experiments

FAPS Benchmark

To evaluate the influence of different alignment templates and the effectiveness of the proposed FAPS, we introduce a well-designed benchmark which includes searching set, training set, validation set and test set. We present our proposed benchmark in Table 1.

The scale of the training dataset is an important factor for face recognition. We separately employ CASIA (Yi et al. 2014) and MS-Celeb-1M (Guo et al. 2016) as middle-scale and large-scale training and searching datasets. For CASIA, we use the full dataset as the searching data and training data. The original MS-Celeb-1M consists of a great number of noisy faces. Therefore, we use MS-Celeb-1M-v1c (Deepglint 2018) which remains the completeness of facial images and is highly clean for training. The MS-Celeb-1M-v1c contains 3.92M images and 86.9K identities, which requires too many computational resources if searching on the full dataset. To reduce the searching time, we sample 30000 identities with 30 images per identity from the whole dataset. This subset is named *Reduced MS-Celeb-1M-v1c*.

Considering different data distributions and characteristics among datasets of the searching set, we enrich the variety of validation set to ensure the generalization of searched policies. The validation set is designed considering the main challenges of face recognition like age, pose and illumination variations. As a result, we build a validation dataset named Cross Challenge in the Wild (CCW), the images are from three datasets in unconstrained environments: LFW(Huang et al. 2008), AgeDB-30(Moschoglou et al. 2017) and CPLFW (Zheng and Deng 2018).

The test set including LFW (Huang et al. 2008), AgeDB-30 (Moschoglou et al. 2017), CALFW (Zheng, Deng, and Hu 2017), CPLFW (Zheng and Deng 2018), MultiPIE (Gross et al. 2010) and IJB-A (Klare et al. 2015). LFW is collected in unconstrained environments with high color jittering and illumination variations. AgeDB-30's primary difficulty lies in the large age gaps and CPLFW has large face pose variations. CALFW demonstrates the age challenge in the wild. MultiPIE is a large multi-view face recognition benchmark. We test our model on subsets of $\pm 90°, \pm 75°, \pm 60°$ yaw angles to evaluate the performance in a large pose situation. The protocol from (Zhou, Cao, and Sun 2018) are followed, where the last 137 subjects with 13 poses, 20 illuminations and neutral expression are selected for testing. We also test the proposed FAPS on IJB-A after training on the large scale dataset MS-Celeb-1M-v1c. Compared with previous datasets, the faces in IJB-A have larger variations and present a more unconstrained scenario. More details of the benchmark are presented in Appendix.

Experimental Settings

We detect the faces by adopting the s3fd detector (Zhang et al. 2017) and localize 68 landmarks via FAN (Bulat and Tzimiropoulos 2017). Images are affined according to the predefined 300×300 average face template T_p as shown in Fig. 3(a). Faces are cropped and resized with different alignment policies for searching, but with consistent policies for training, validation and testing. The cropped faces are then resized to 112×112. All image pixel values are subtracted with the mean 127.5 and divided by 128. During training, horizontally flipping with probability 0.5 are used as the data augmentation.

The widely used ResNets (He et al. 2016) with embedding structure (Deng et al. 2019) are employed as our recognition networks. The embedding dimension is set to 512. To accelerate the searching process, ResNet18 is adopted as the searching network. ResNet50 is used to train on the training set. ArcFace (Deng et al. 2019) is served as the loss function during searching and training. We implement FAPS with PyTorch (Paszke et al. 2017) and Ray Tune (Moritz et al. 2018).

During searching, the population size of models *N* is set to 8. Cosine annealing learning rate that decays from 0.1 to 0.00001 is applied as LR-scheduler to smooth the process. The momentum is set to 0 to eliminate the impact of the input changes. The crop size m_{max} and m_{min} are set to 232 and 160, respectively. The vertical shift δ_{max} and δ_{min} are 24 and -32. We set the magnitude parameter of crop size $s_m = 8$ and the magnitude parameter of vertical shift

Table 1: FAPS Benchmark

Benchmark	CASIA	MS-Celeb-1M-v1c
Searching Set	CASIA	*Reduced MS-Celeb-1M-v1c*
Training Set	CASIA	MS-Celeb-1M-v1c
Validation Set	CCW	CCW
Test Set	LFW AgeDB-30 CPLFW CALFW MultiPIE	LFW AgeDB-30 CPLFW CALFW MultiPIE IJB-A

$s_\delta = 4$. Under this setting, we have 93 candidates in the template searching space $\mathcal{P}$. More setting details are shown in Appendix.

Compared Methods

For comparison, we map the widely-used 5-points template presented in ArcFace (Deng et al. 2019) to the predefined 300×300 template T_p, which results in policy $\boldsymbol{p} = \{190, -7\}$. Another 25-points alignment template utilized by MFR (Guo et al. 2020b) and works (Zhu et al. 2019; Guo et al. 2018) is mapped to $\{198, -15\}$. We call policy $\{m_{min}, 0\} = \{160, 0\}$ the *TightROI* which involves few external face features. *SuperROI* as well as the aforementioned three policies are treated as compared policies. We further compare the proposed FAPS with the spatial-transform based methods ReST (Wu et al. 2017) and GridFace(Zhou, Cao, and Sun 2018).

Fig. 6 shows some aligned faces with different policies. Templates used by ArcFace and MFR drops part of the chin and includes almost all the forehead, while MFR's contains a bit more external features than ArcFace's. The *SuperROI* contains all the facial features and *TightROI* drops half of the chin and forehead and focuses on the internal facial features. ReST and GridFace coupled alignment with recognition network, they can hardly be mapped into our search space.

Searching on CASIA

In this section, CASIA is used as the searching and training sets. The corresponding validation/test sets are presented in Table 1. FAPS's searching process takes **131** GPU hours with 8 Tesla V100 GPUs. As a comparison, the grid search method with ResNet18 takes about **9102** GPU hours. With the searched alignment policy, we train the ResNet50 from scratch for 32 epochs. The learning rate is initialized by 0.1 and divided by 10 at epoch 20 and 28.

Results are summarized in Table 2 and Table 3 (results of the baseline will be discussed in Ablation Study). We denote the searched alignment policy FAPS$_C$(192,4). Obviously, FAPS$_C$(192,4) surpasses the compared policies on all test datasets. For example, on LFW, FAPS$_C$(192,4) outperforms all other policies, especially the *TigthROI*. With the same training dataset, FAPS$_C$(192,4) achieves a 0.45% improvement above ReST. On AgeDB-30 and CALFW, FAPS$_C$(192,4) shows significant improvements over the best results from compared policies by 0.78% and 1.15%. As shown in Fig. 6, FAPS$_C$(192,4) drops more hair than ArcFace's and MFR's but remains more chin. This indicates that hair is not helpful for face recognition with age challenge as people's hairstyles usually change during their lifetime, while the chin and the outline of chin remain unchanged.

Table 2: Verification performance (%) at different alignment policies with ResNet50 backbone. MS1M: MS-Celeb-1M-v1c. FAPS$_C$(192,4) and FAPS$_M$(200,4) denote the policies searched on CASIA and *Reduced MS-Celeb-1M-v1c*, respectively.

Training Set	Method	LFW	AgeDB-30	CALFW	CPLFW
CASIA	ReST	99.03	-	-	-
	ArcFace (190,-7)	99.43	94.42	90.92	85.15
	MFR (198,-15)	99.43	94.47	91.15	84.75
	TigthROI (160,0)	99.17	94.23	91.15	85.07
	SuperROI (232,0)	99.43	94.47	90.48	83.97
	baseline (184,4)	99.45	95.03	91.07	**85.88**
	FAPS$_C$ (192,4)	**99.48**	**95.25**	**92.07**	85.43
MS1M	GridFace	99.70	-	-	-
	ArcFace (190,-7)	99.72	98.02	95.23	87.98
	MFR (198,-15)	99.77	97.78	95.47	87.28
	TigthROI (160,0)	99.73	97.95	95.47	88.13
	SuperROI (232,0)	99.77	**98.25**	95.47	88.05
	FAPS$_C$ (192,4)	99.78	98.10	**95.78**	88.12
	FAPS$_M$ (200,4)	**99.82**	98.08	95.65	**88.95**

For profile faces, FAPS$_C$(192,4) gains improvement over other compared policies on CPLFW, MultiPIE $\pm 75°$ and MultiPIE $\pm 90°$. On the less challenging MultiPIE $\pm 60°$, FAPS$_C$(192,4) performs as good as MFR and *TightROI*. These results show FAPS's searched alignment policy gains superiority over handcrafted ones for faces with large pose variations. This mainly because profile faces are aligned to one side of the images (as shown in Fig. 6). Policies with too small crop sizes (*e.g.*, *TightROI*) filter out useful face features, while large crop sizes (*e.g.*, *SuperROI*) can bring irrelevant features and background noise. In contrast, our FAPS can find a trade-off and therefore can focus on key features.

Figure 6: Face images aligned with different templates. The first two rows show faces of the same person in CALFW. Faces of the last two rows are from MultiPIE $\pm 90°$ subset and MultiPIE $0°$ subset respectively, they are the same identity as well.

Searching on MS-Celeb-1M-v1c

In this section, *reduced MS-Celeb-1M-v1c* is used as the searching data. The searching process takes **234** GPU hours with 8 Tesla V100 GPUs, while the grid search method takes more than **4812** GPU hours. After the searching phase, we train ResNet50 for 16 epochs from scratch with the searched alignment policy on the full data, with learning rate initialized as 0.1 and dropped by 10 at the 8th and 14th epochs.

Results are showed in Table 2, 3, 4. When compared with other handcrafted alignment policies, FAPS's searched policy on *Reduced MS-Celeb-1M-v1c* (FAPS$_M$(200,4)) outperforms other policies on almost all the datasets. On LFW, FAPS$_M$(200,4) outperforms the human-designed policies by at least 0.05%. As the verification accuracy on LFW is almost saturated around 99.80%, the improvement is non-

Table 3: Rank-1 recognition rates (%) for different poses at different alignment policies on MultiPIE with ResNet50 backbone. FAPS$_C$(192,4) and FAPS$_M$(200,4) denote the policies searched on CASIA and *Reduced MS-Celeb-1M-v1c*, respectively.

Training Set	Method	$\pm 90°$	$\pm 75°$	$\pm 60°$
CASIA	ArcFace (190,-7)	89.5	97.0	99.3
	MFR (198,-15)	91.2	97.7	**99.7**
	TigthROI (160,0)	90.8	97.6	**99.7**
	SuperROI (232,0)	90.7	97.1	99.3
	baseline (184,4)	90.4	97.5	99.6
	FAPS$_C$ (192,4)	**91.7**	**98.3**	**99.7**
MS1M	GridFace	75.4	94.7	99.2
	ArcFace (190,-7)	70.4	98.8	**100.0**
	MFR (198,-15)	71.9	98.9	**100.0**
	TigthROI (160,0)	68.7	98.4	**100.0**
	SuperROI (232,0)	70.7	98.0	99.9
	FAPS$_C$ (192,4)	74.6	**99.0**	**100.0**
	FAPS$_M$ (200,4)	**76.6**	98.8	**100.0**

negligible. On CALFW, FAPS$_M$(200,4) outperforms other handcrafted alignment policies by almost 0.2%. For profile faces, the searched policy FAPS$_M$(200,4) can obviously boost the performance on both CPLFW and MultiPIE $\pm 90°$ by 0.82% and 4.7%, respectively. On the challenging dataset IJB-A, FAPS$_M$(200,4) achieves best verification and identification performance. The verification accuracy with FAR at 0.001 improves other manual templates by more than 2.6%.

With the same training dataset MS1M, Our FAPS$_M$(200,4) achieves a 0.12% improvement above GridFace on LFW. On MultiPIE $\pm 90°$, $\pm 75°$ and $\pm 60°$, FAPS$_M$(200,4) outperforms GridFace by clear margins. On IJB-A, FAPS$_M$(200,4) gains obvious improvement on verification accuracy(3.0% and 7.3%), it also shows better performance on verification accuracy.

To further verify the generalization of our searched template, we train ResNet50 on MS-Celeb-1M-v1c with the policy FAPS$_C$(192,4) searched on the smaller dataset CASIA. When compared with handcrafted alignment policies, FAPS$_C$(192,4) also gains better performance on almost all the datasets while a little bit inferior to FAPS$_M$(200,4)'s. It shows improvements on LFW, CALFW, MultiPIE $\pm 90°$, $\pm 75°$ and gains comparable performance on CPLFW and MultiPIE $\pm 60°$. On IJB-A, FAPS$_C$(192,4) boosts the verification accuracy with FAR at 0.001 and the Rank-1 accuracy. These results show the generalization of the searched alignment policies of FAPS. Once searched on one dataset, the searched policy can further improve the recognition performance when trained on different datasets.

On both CASIA and MS-Celeb-1M-v1c, the searched alignment policies gain better performance. It shows that compared to current human-designed alignment templates, the optimal one can be searched by FAPS to facilitate the face recognition performance. The searched alignment policy can also generalize across different training datasets. Moreover, although the searched alignment policy of MS-Celeb-1M-v1c is different from CASIA's, the input facial area decided by the two searched policies are almost overlapped (IOU 0.92). Almost all chin and part of the forehead are contained for both policies. The results show that adding proper external facial features is beneficial to recognition.

Ablation Study

Effectiveness of Intersection based Crossover We first evaluate *Intersection based Crossover*, the method we proposed to facilitate the discovering of better alignment policies. To analyze its impact, we search for the CASIA's alignment policy under the same setting as that in section Searching on CASIA, but without *Intersection based Crossover*. The searched alignment policy without *Intersection based Crossover* is named baseline. The results are summarized in Table 2 and 3. The policy FAPS$_C$(192,4) discovered with *Intersection based Crossover* shows better results compared to the baseline at almost all test datasets. Specifically, FAPS$_C$(192,4) outperforms baseline by 1.0% at CALFW, 1.3% and 0.8% at MultiPIE $\pm 90°$ and $\pm 75°$. At CPLFW, FAPS$_C$(192,4) is slightly inferior to baseline. The reason may be that CPLFW contains more complex background and occlusion than MultiPIE. The facial area decided by FAPS$_C$(192,4) is a bit lager than baseline's, which means more background noise involved.

Table 4: Results on IJB-A with searched policies FAPS$_C$(192,4) and FAPS$_M$(200,4). The training set is MS-Celeb-1M-v1c.

Method ↓	Verification		Identification	
Metric →	@FAR = 0.01	@FAR = 0.001	@Rank-1	@Rank-5
GridFace	92.1 ± 0.8	83.9 ± 1.4	92.9 ± 1.0	96.2 ± 0.5
ArcFace (190,-7)	94.5 ± 0.6	87.1 ± 1.4	93.1 ± 0.8	95.5 ± 0.4
MFR (198,-15)	94.7 ± 0.6	88.6 ± 1.0	93.7 ± 0.7	96.0 ± 0.6
TigthROI (160,0)	93.6 ± 0.8	82.1 ± 2.8	92.4 ± 0.7	95.0 ± 0.6
SuperROI (232,0)	95.1 ± 0.7	87.4 ± 1.9	93.7 ± 0.8	95.8 ± 0.5
FAPS$_C$ (192,4)	94.8 ± 0.6	89.7 ± 1.4	93.8 ± 0.8	95.9 ± 0.5
FAPS$_M$ (200,4)	$\mathbf{95.1 \pm 0.6}$	$\mathbf{91.2 \pm 0.6}$	$\mathbf{94.1 \pm 0.7}$	$\mathbf{96.4 \pm 0.4}$

Performance consistency between ResNet18 and ResNet50 To prove the generalization (*i.e.*, not biased to a specific network) of our searched policy, we further evaluate the performance of FAPS$_C$ (192,4) on ResNet18. We train ResNet18 on CASIA using the policy FAPS$_C$(192,4) with the same settings as ResNet50 in Table 2 and present the results on Table 5. The performances of the compared policies are also trained with ResNet18 on CASIA. From both Table 2 and Table 5, we conclude that compared with the other policies, FAPS$_C$ (192,4) can improve the performances of different backbones at all datasets, which reveals the performance consistency of our searched policy.

Table 5: Verification performance (%) with different alignment policies on ResNet18.

Alignment Policy	LFW	AgeDB-30	CALFW	CPLFW
ArcFace (190,-7)	99.10	93.18	89.05	78.43
MFR (198,-15)	99.12	93.30	89.45	79.22
TigthROI (160,0)	99.02	93.73	88.78	79.30
SuperROI (232,0)	99.18	93.38	88.80	79.22
FAPS$_C$ (192,4)	**99.20**	**94.02**	**89.47**	**80.28**

Conclusions

In this paper, we explore the effects of different alignment templates on face recognition and propose a fast and effective alignment policy search method named FAPS. The searched templates via FAPS achieve better recognition performance compared to human-designed ones on multiple test datasets and generalize across different training datasets. Besides, our searched templates reveal that except for the internal facial features like eyes, nose and mouth, external features like chin and jawline are helpful for face recognition. This also sheds some light on the further development of face recognition.

Acknowledgments

This work was supported in part by the National Key Research & Development Program (No. 2020AAA0140002), Chinese National Natural Science Foundation Projects #61876178, #61806196, #61976229.

References

Andrews, T. J.; Davies-Thompson, J.; Kingstone, A.; and Young, A. W. 2010. Internal and External Features of the Face Are Represented Holistically in Face-Selective Regions of Visual Cortex. *Journal of Neuroscience the Official Journal of the Society for Neuroscience* 30(9): 3544–3552.

Bergstra, J.; and Bengio, Y. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research* 13(Feb): 281–305.

Bulat, A.; and Tzimiropoulos, G. 2017. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In *International Conference on Computer Vision*.

Chen, Y.; Yang, T.; Zhang, X.; Meng, G.; Xiao, X.; and Sun, J. 2019. DetNAS: Backbone search for object detection. In *Advances in Neural Information Processing Systems*, 6638–6648.

Chu, X.; Zhang, B.; Xu, R.; and Li, J. 2019. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. *arXiv preprint arXiv:1907.01845* .

Cubuk, D. E.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, V. Q. 2019. AutoAugment - Learning Augmentation Strategies From Data. *CVPR* 113–123.

Deepglint. 2018. http://trillionpairs.deepglint.com/overview.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gross, R.; Matthews, I.; Cohn, J.; Kanade, T.; and Baker, S. 2010. Multi-pie. *Image and Vision Computing* 28(5): 807–813.

Guo, J.; Zhu, X.; Lei, Z.; and Li, S. Z. 2018. Face synthesis for eyeglass-robust face recognition. In *Chinese Conference on Biometric Recognition*, 275–284. Springer.

Guo, J.; Zhu, X.; Yang, Y.; Yang, F.; Lei, Z.; and Li, S. Z. 2020a. Towards Fast, Accurate and Stable 3D Dense Face Alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Guo, J.; Zhu, X.; Zhao, C.; Cao, D.; Lei, Z.; and Li, S. Z. 2020b. Learning Meta Face Recognition in Unseen Domains. *arXiv preprint arXiv:2003.07733* .

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 *European conference on computer vision*, 87–102. Springer.

Guo, Z.; Zhang, X.; Mu, H.; Heng, W.; Liu, Z.; Wei, Y.; and Sun, J. 2019. Single path one-shot neural architecture search with uniform sampling. *arXiv preprint arXiv:1904.00420* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ho, D.; Liang, E.; Stoica, I.; Abbeel, P.; and Chen, X. 2019. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. *international conference on machine learning* .

Huang, G. B.; Mattar, M.; Berg, T.; and Learned-Miller, E. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W. M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; Fernando, C.; and Kavukcuoglu, K. 2017. Population Based Training of Neural Networks .

Kang, B.-N.; Kim, Y.; Jun, B.; and Kim, D. 2019. Attentional Feature-Pair Relation Networks for Accurate Face Recognition. *arXiv preprint arXiv:1908.06255* .

Klare, B. F.; Klein, B.; Taborsky, E.; Blanton, A.; Cheney, J.; Allen, K.; Grother, P.; Mah, A.; and Jain, A. K. 2015. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1931–1939.

Lapedriza, A.; Masip, D.; and Vitria, J. 2005. Are external face features useful for automatic face classification? In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, 151–151. IEEE.

Lim, S.; Kim, I.; Kim, T.; Kim, C.; and Kim, S. 2019. Fast AutoAugment. *NeurIPS* 6662–6672.

Liu, J.; Deng, Y.; Bai, T.; and Huang, C. 2015. Targeting Ultimate Accuracy: Face Recognition via Deep Embedding. *CoRR* abs/1506.07310.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M. I.; et al. 2018. Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 561–577.

Moschoglou, S.; Papaioannou, A.; Sagonas, C.; Deng, J.; Kotsia, I.; and Zafeiriou, S. 2017. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 51–59.

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 .

Ryuichiro, H.; Jan, Z.; Kazuki, Y.; and Hideki, N. 2019. Faster AutoAugment: Learning Augmentation Strategies using Backpropagation .

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Spears, W. M. 1993. Crossover or mutation? In *Foundations of genetic algorithms*, volume 2, 221–237. Elsevier.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. *Advances in Neural Information Processing Systems* 3(January): 1988–1996. ISSN 10495258.

Sun, Y.; Wang, X.; and Tang, X. 2014a. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1891–1898.

Sun, Y.; Wang, X.; and Tang, X. 2014b. Deep learning face representation from predicting 10,000 classes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1891–1898. ISSN 10636919.

Sun, Y.; Wang, X.; and Tang, X. 2014c. Deeply learned face representations are sparse, selective, and robust. *CoRR* abs/1412.1265. URL http://arxiv.org/abs/1412.1265.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014a. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014b. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 1041–1049.

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. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 5265–5274. ISSN 10636919.

Wu, W.; Kan, M.; Liu, X.; Yang, Y.; Shan, S.; and Chen, X. 2017. Recursive spatial transformer (rest) for alignment-free face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 3772–3780.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923* .

Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; and Li, S. Z. 2017. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, 192–201.

Zhang, X.; Wang, Q.; Zhang, J.; and Zhong, Z. 2020. Adversarial AutoAugment. *International Conference on Learning Representations* .

Zheng, T.; and Deng, W. 2018. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep* 5.

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

Zhou, E.; Cao, Z.; and Sun, J. 2018. Gridface: Face rectification via learning local homography transformations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Zhu, X.; Liu, H.; Lei, Z.; Shi, H.; Yang, F.; Yi, D.; Qi, G.; and Li, S. Z. 2019. Large-scale bisample learning on id versus spot face recognition. *International Journal of Computer Vision* 127(6-7): 684–700.