Pooling Faces: Template based Face Recognition with Pooled Face Images

Tal Hassner[1,2] Iacopo Masi[3] Jungyeon Kim[3] Jongmoo Choi[3] Shai Harel[2]
Prem Natarajan[1] Gérard Medioni[3]

[1] Information Sciences Institute, USC, CA, USA

[2] The Open University of Israel, Israel

[3] Institute for Robotics and Intelligent Systems, USC, CA, USA

Abstract

We propose a novel approach to template based face recognition. Our dual goal is to both increase recognition accuracy and reduce the computational and storage costs of template matching. To do this, we leverage on an approach which was proven effective in many other domains, but, to our knowledge, never fully explored for face images: average pooling of face photos. We show how (and why!) the space of a template's images can be partitioned and then pooled based on image quality and head pose and the effect this has on accuracy and template size. We perform extensive tests on the IJB-A and Janus CS2 template based face identification and verification benchmarks. These show that not only does our approach outperform published state of the art despite requiring far fewer cross template comparisons, but also, surprisingly, that image pooling performs on par with deep feature pooling.

1. Introduction

Template based face recognition problems assume that both probe and gallery items are potentially represented using *multiple* visual items rather than just one. Unlike the term *set based* face recognition, *template* was adopted by the recent Janus benchmarks [26] to emphasize that templates may have heterogeneous content (e.g., images, videos) contrary to older benchmarks such as the YouTube Faces (YTF) [43] in which sets contained images of a single nature (e.g., video frames). The template setting was designed to reflect many real-world biometric scenarios, where capturing a subject's facial appearance is possible more than once and using different acquisition methods.

Ostensibly, having many images instead of one provides more appearance information which in turn should lead to more accurate recognition. In reality, however, this is not always the case. The real-world images populating these templates vary greatly in quality, pose, expression and more. Matching across templates requires that all these issues are taken under consideration to avoid skewing matching scores based on these and other confounding factors. Doing this well requires knowing which images should be compared and how to weigh the similarities of different cross-template image pairs. Beyond these, however, are also questions of complexity: How should two templates be efficiently compared without compromising (or even gaining) accuracy?

Previous work on this problem focused on the set based setting, often with the YTF benchmark, and proposed various set representations and set-to-set similarity measures. These prescribe representing face sets as anything from linear subspaces (e.g., [11, 18]) to non-linear manifolds [9, 32]. More recent template based methods, however, seem to prefer explicitly storing all face images over using more specialized set representations [1, 7, 34, 35, 38]. Set similarity is then computed by measuring the similarities between all cross template image pairs and aggregating them into a single, template based similarity score.

We propose simple image averages (a.k.a., average pooled faces, a.k.a., 1st order set statistics) as template representations. Pooling images using pixel-wise average or median is long since known to be an effective means of correcting images, removing noise and overcoming incidental occlusions (e.g., the seminal work of [20, 21, 22]). Very recently, *feature* pooling (rather than pooling image intensities) was proposed as an extremely useful approach for endowing existing features with invariant properties. Two such examples are scale invariance by multi-scale pooling of SIFT features [31] in [10] and pose (viewpoint) invariance by cross-pose pooling of deep features [39].

Rather than feature pooling, we return to pooling images directly. As we discuss in Sec. 3, previous work avoided relying only on this representation for face image sets and we explain why this was so. We show that the underlying requirement of successful image based pooling methods – image alignment – can easily be satisfied by 3D alignment techniques such as face *frontalization* [14]. Moreover, using a number of technical novelties and careful partitioning of the images in a template, based on head pose and image quality, we show that few pooled images capture facial appearances better than the original template. That is, we provide improved template matching scores but require fewer images to represent templates.

We test performance on the Janus CS2 and IJB-A datasets, using deep feature representations to encode our pooled images. We show that both face verification and identification results outperform recent state of the art. Finally, we compare our *image* pooling to the increasingly popular approach of deep *feature* pooling. Surprisingly, our results show that pooled images perform on par with pooled features, despite the fact that image alignment and averaging is computationally cheaper than deep feature extraction.

2. Related work

Much of the relevant work done in the past focused on the set based settings, where probe and gallery items typically comprised of multiple frames from the same video. Possibly the simplest approach to representing and matching image sets is to store the images of each set (or features extracted from them) directly, and then measure the distance between two sets by aggregating the distances between all cross-set image pairs (e.g., *min-dist* [43]). Other, more elaborate methods designed for this purpose can broadly be categorized as belonging to four different categories.

Set *Convex* or *Affine hulls* were both proposed as representations of face image sets. Convex hull was used by [5] and then extended to the use of Affine hull in [15]. These methods are most effective when many images are available in each set and these hulls are well defined.

Subspace methods represent sets using linear subspaces [3, 4, 11, 18, 24]. Though the underlying assumption that all set elements lie close to a linear subspace may seem restrictive, it provides a computationally efficient representation and a natural definition for set-to-set distances: the angles between different subspaces [4]. Real world photos of faces, however, rarely lie on linear subspaces. Using such subspaces to represent them risks substantial loss of information and a degradation in recognition capabilities.

When set items cannot be assumed to reside on a linear subspace, sets may still be represented by *non-linear manifolds*. Some examples of this approach include [9, 19, 32, 42]. These typically require manifold learning techniques and manifold-to-manifold distance definitions which can be expensive to compute in practice.

Finally, various *distribution based* representations were also considered for this purpose. Possibly the most widely used are histogram representations such as the bag of features [28], Fisher vectors [36] and Vector of Locally Aggregated Descriptors (VLAD) [23]. These are typically applied to sets of local descriptors, rather than images. Sets containing entire face photos were represented by 1st to n'th order statistics in [33]. Alternatively, by assuming that sets of faces are Gaussians, they were represented using covariance matrices (2nd order statistics) in [41] and [45].

3. Motivation: Are 1st order statistics enough?

Let a (gallery or probe) face template be represented by the set of its member images (assuming that videos are represented by their individual frames), as: $\mathcal{F} = \{\mathbf{I}_1, ..., \mathbf{I}_N\}$. where $\mathbf{I}_i \in \mathbb{R}^{n\times m\times 3}$ are RGB images, aligned by cropping the bounding box centered on the face and rescaling it to the same dimensions for all images (i.e., images are assumed to be aligned for translation and scale). The 1st order statistics of this set (the average pooled face) is simply defined as:

$$\mathbf{F} \doteq avg(\mathcal{F}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}_i \qquad (1)$$

Although some of the methods surveyed in Sec. 2 used 1st order statistics of face sets as part of their representations, none ventured so far as to propose using them alone, and for good reason: High order statistics and/or metric learning are required to represent and match facial appearance variations that cannot be captured effectively only by 1st order statistics. Fig. 1 illustrates this by showing face images from a single template and their average. Apparently, averaging loses much of the information available in each individual image in favor of noise.

Also evident in Fig. 1 is that at least to some extent this is an alignment problem: if faces appear in exactly the same alignment (in particular, the same head pose), their average is far clearer. This was recently demonstrated in [14] which showed that better head pose alignments produce sharper average images.

We go beyond the work in [14] and propose to cancel out variations in pose and image quality, in order to produce superior pooled faces which can be used for recognition. This, as an alternative to using high order statistics to represent face sets or expensive metric learning schemes to match them.

Specifically, we partition a set of images into subsets containing faces which share similar appearances. We further reduce facial appearances by 3D head pose alignment. As a consequence, a face set is represented by a small collection of 1st order statistics, extracted from few subsets of the original template. Doing so has a number of attractive advantages over previous work:

- **Reduced computational costs.** Image alignment and averaging are computationally cheaper than other existing representations.

- **Faster matching.** Matching two templates is also quite efficient, due to the drop in the number of images representing each set. Moreover, this approach does not require expensive metric learning schemes to address appearance variations.

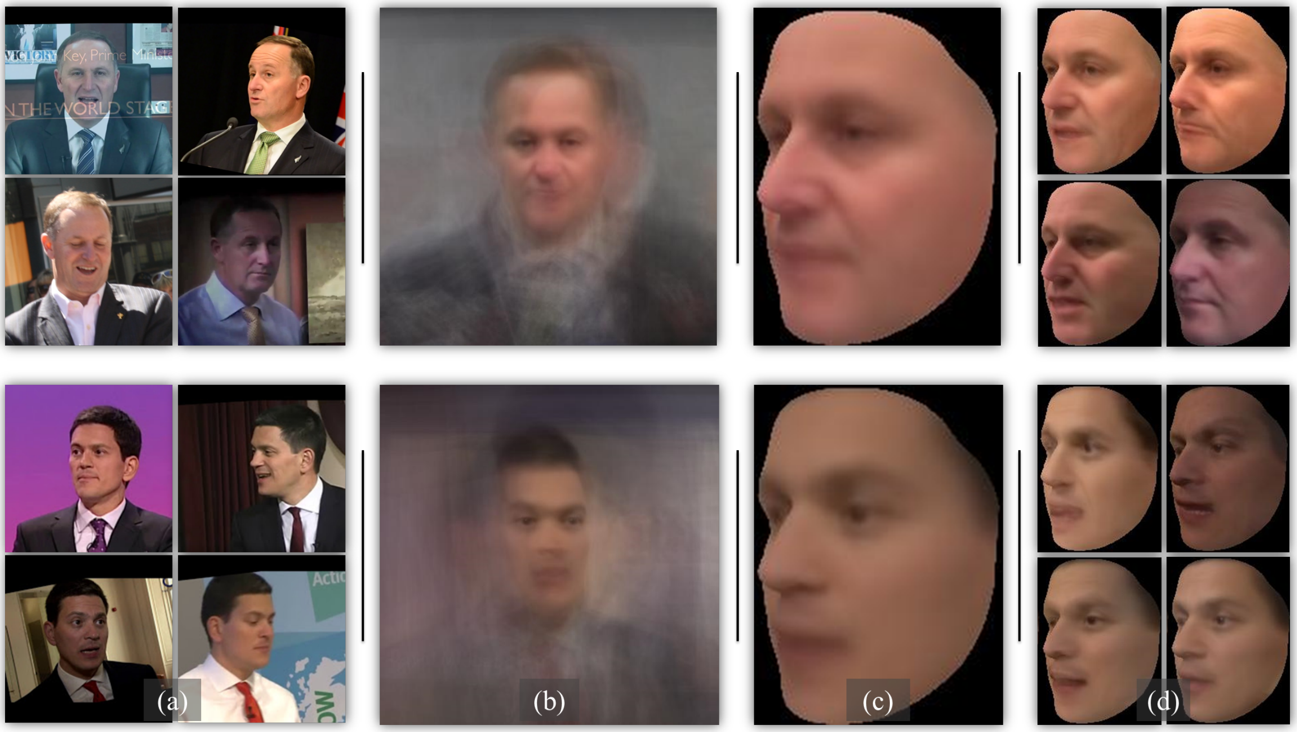


Figure 1. **Pooled faces**. (a) Example images from Janus [26] templates. (b) Averages of all in-plane aligned template images. The subjects are hardly recognizable in these averages. (c) Averages of all 3D aligned template images. Though better than (b), these are over smoothed and still hard to recognize. (d) Averages of 3D aligned images from four different face bins. These retain more high frequency information and details necessary for recognizing the subjects in the photos.

- **Improved accuracy.** Despite reduced storage and computational costs, accuracy actually improves. This is likely due to the known properties of average images to reduce noise and remove incidental occlusions.

4. Face pooling

Our pipeline is illustrated in Fig. 2. Given a face template $\mathcal{F}$, we align its images in 3D and then bin the aligned images according to pose and image quality. Images falling into the same bin are pooled, Eq. (1), and the pooled images are encoded using a convolutional neural network (CNN). Finally, we use these CNN features to match templates. We next describe these steps in detail.

4.1. Binning by head pose

3D head pose estimation: The recent work of [13] showed that the 6dof pose of a head appearing in a 2D image can be estimated by minimizing the geometrical distances between extracted 2D facial landmarks and their corresponding reprojected 3D landmarks on a generic 3D face model. In this work, we perform a similar process, with slight changes.

Given a bounding box around a face, we detect 68 landmarks using CLNF [2]. Bounding boxes were estimated using the DLIB library of [25]. We used CLNF to detect the same landmarks in a rendered image of a generic 3D face. The correspondences between the 3D coordinates on the generic model and its rendered view are obtained using the rendering code of [13]. Hence, given the detected points $\mathbf{p}_{i,i=1..68} \in \mathbb{R}^2$ on the input photo, and their corresponding points $\hat{\mathbf{p}}_{i,i=1..68} \in \mathbb{R}^2$ on the rendered view, we have the 3D coordinates for these points, $\hat{\mathbf{P}}_{i,i=1..68} \in \mathbb{R}^3$, on the generic face model.

Assuming the principal point is in the image center we use the 68 correspondences $(\mathbf{p}_i, \hat{\mathbf{P}}_i)$ to solve for the extrinsic camera parameters with the PnP method [12]. This provides us with a camera matrix $\mathbf{M} = \mathbf{K}\ [\mathbf{R}\,\mathbf{t}]$ minimizing the projection errors of the 3D landmarks to the landmarks detected on the input photo. The estimated pose $\mathbf{M}$ is then decomposed to provide a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ for the yaw, pitch and roll angles of the head.

These three angles are used in three ways: roll compensation, head pose quantization and pose cancellation. Roll compensation simply means in-plane alignment of the faces so that the line between the eyes is horizontal [12].

Head pose quantization: Once 2D roll is eliminated, we consider only yaw angles (in practice we found pitch variations in our datasets to be small, and so only yaw angle variations were addressed; pitch angles can presumably

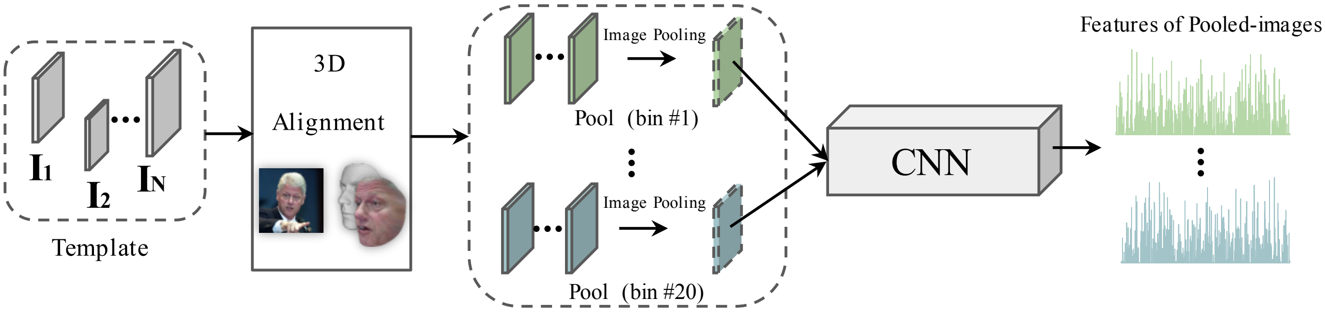


Figure 2. **Template representation and matching pipeline**. Illustrates the various stages of our approach. Please see text for details.

also be used to quantize head poses to, e.g., ±15° pitch angle bins). Yaw angles ($|\theta|$) are quantized into four groups, $\{(0° \leq |\theta| < 20°), (20° \leq |\theta| < 40°), (40° \leq |\theta| < 60°), (60° \leq |\theta|\}$.

Head pose cancellation: All images in all bins are then aligned in 3D to remove any remaining pose variations. We perform a process similar to the one described in [14], including soft-symmetry. Unlike [14], our own rendering code produced faces over a black background (the original background of the input photos was not preserved in rendering). More importantly, we found that better overall performance is obtained by rendering the faces not to frontal pose, as in [14], but to a 40° (half-profile) view (Fig. 1).

4.2. Binning by image quality

Inspecting the images available in the templates of a recent collection such as Janus [26] reveals that their quality varies significantly from one photo to another. This may be due to motion blur, difficult viewing conditions or low quality camera gear. Regardless of the reason, one immediate consequence of this is that pooling low quality photos may degrade the average image.

One way to address this is to eliminate poor images. We found, however, that doing so reduces overall accuracy, presumably because even poor photos carry some valuable information. As a consequence, these images are still used, but are pooled separately from high quality images.

Estimating Facial Image Quality (FIQ): We seek a FIQ measure which assigned a normalized image quality based score for a facial image **I**. Work on estimating image quality is abundant. We tested several existing methods for image quality estimation, ultimately choosing the *Spatial-Spectral Entropy based Quality* (SSEQ) [29].

SSEQ is a *no-reference* image quality assessment model that utilizes local spatial and spectral entropy features on distorted images [29]. It uses a support vector machine (SVM) trained to classify image distortion and quality. A key advantage of SSEQ is that its final index allows assessing the quality of a distorted photo across multiple distortion categories. It additionally matches well with human subjective opinions of image quality [29].

We use the code originally released for SSEQ by its authors [30]. Normalized SSEQ scores were partitioned into five image quality bins, with bin limits determined empirically. Specifically, given face image the following threshold values are used to assign the image with a quality index:

$$Q(\mathbf{I}) = \begin{cases} 0, & \text{if } -\infty < SSEQ(\mathbf{I}) < 0.45 \\ 1, & \text{if } 0.45 \leq SSEQ(\mathbf{I}) < 0.55 \\ 2, & \text{if } 0.55 \leq SSEQ(\mathbf{I}) < 0.65 \\ 3, & \text{if } 0.65 \leq SSEQ(\mathbf{I}) < 0.75 \\ 4, & \text{if } 0.75 \leq SSEQ(\mathbf{I}) < \infty, \end{cases} \tag{2}$$

where $SSEQ(\mathbf{I})$ is the FIQ measure of the input image **I**.

4.3. Representing and comparing templates

Template representation: Table 1 summarizes the bin indices used in this work. All told, 20 bins are available, though, as we later show, typical templates have far fewer bins populated by images in practice.

Pooled images are represented using deep features. Specifically, we use the VGG-19 CNN of [6] to encode face images. This 19 layer network was originally trained on the large scale image recognition benchmark (ILSVRC) [37]. We fine tune the weights of this network twice: first on original CASIA WebFace images [44], with the goal of learning to recognize 10,575 subject labels of that set.

A second fine tuning is performed again using CASIA images. This time, however, training is performed following image pooling, using the process described in Sec. 4. Since CASIA has not template definitions, we use subject labels instead: We take random subsets of images from the

Pose (head yaw)	[0°...20°), [20°...40°), [40°...60°), [60°...90°]
Quality (SSEQ [29])	$(-\infty$...0.45), [0.45...0.55), [0.55...0.65), [0.65...0.75), [0.75...∞)

Table 1. **Bin indices.** The quantized pose and image quality in our template representation. Twenty bins used altogether, though few are populated in practice.

same subject. Each subset is then treated as a template. Subjects for which only a single image exists are used in this second fine tuning step, without pooling.

Matching images and templates: A trained CNN is defined by linear functions and non-linear activations. The network is parametrized with a set of convolutional layers and fully connected layers, including values for the weights, $\mathbf{W}$, and biases, $\mathbf{b}$. The CNN is used to extract feature representations, $\mathbf{x} = f(\mathbf{I}; \{\mathbf{W}, \mathbf{b}\})$, for each image, $\mathbf{I}$ (pooled or not). We take the response produced after the fully connected layer fc7 as the image representation. Given an image $\mathbf{I}_p$ in a probe template $\mathcal{P}$ and $\mathbf{I}_g$ in a gallery template $\mathcal{G}$, we compute their similarity, $s(\mathbf{x}_p^{\text{fc7}}, \mathbf{x}_g^{\text{fc7}})$, by taking the normalized cross correlation (NCC) of their feature vectors.

A *template similarity* is defined by aggregating these scores for all cross-template (pooled) image pairs (i.e., all-vs-all matching of features extracted from pooled images). We define the similarity $s(\mathcal{P}, \mathcal{G})$ of two templates $\mathcal{P}$ and $\mathcal{G}$ as follows. After computing all pair-wise pooled-image level similarity scores, these values are fused using SoftMax: $s_\beta(\mathcal{P}, \mathcal{G})$, defined in Eq.(3), below. The use of SoftMax here to aggregate image level similarity scores can be considered a weighted average which depends on the score to set the weights as:

$$s_\beta(\mathcal{P}, \mathcal{G}) = \frac{\sum_{p \in \mathcal{P}, g \in \mathcal{G}} \; w_{pg} \; s(\mathbf{x}_p, \mathbf{x}_g)}{\sum_{p \in \mathcal{P}, g \in \mathcal{G}} \; w_{pg}}, \quad w_{pg} \doteq e^{\beta \; s(\mathbf{x}_p, \mathbf{x}_g)} \tag{3}$$

As the final template similarity score, we average the SoftMax responses over multiple values of $\beta = [0...20]$.

5. Experiments

We tested our pooling approach extensively on the IARPA Janus Benchmark-A (IJB-A) and JANUS CS2 benchmark. Compared to previous benchmarks (e.g., Labeled Faces in the Wild [16, 17] and YTF [43]) this dataset is far more challenging and diverse in its contents. In particular IJB-A brings two design novelties over these older benchmarks:

- Janus faces reflect a wider range of challenges, including extreme poses and expressions, low quality and noisy images and occlusions. This is mainly due to the its design principles which emphasize heterogeneous media collections.

- Subjects are represented by templates of images from multiple sources, rather than single images. They are moreover described by both still-images and frames from multiple videos. Throughout this paper, we therefore followed their terminology, referring to image templates rather than sets (as in the YTF collection).

Fig. 3 provides a qualitative comparison between LFW and IJB-A images, highlighting the difference between a subject included in both sets: images in LFW are strongly biased towards web based production quality images, whereas IJB-A images are of poorer quality and wide pose changes.

5.1. Performance Metrics

Standard Janus verification metrics: We report the standard performance metrics for IJB-A. For both the verification and identification protocols we show different recall values (True Positive Rate) at different cut-off points in the False Positive Rate (FPR) of the ROC. The FPR is sampled at an order of magnitude less each time, ranging from TPR-1%F (TPR at 1% of FPR) to the most difficult point at TPR-0.01%F (TPR at 0.01% of FPR).

This evaluation fits perfectly with the face verification protocol defined in IJB-A verification, as also previously done for LFW. Considering the ROC, we also show the opposite, which we believe to be more relevant in real-world scenarios: assume that we want to have a fixed recall of 85%, the system should report what is the FPR. We denoted this metric as FPR-85T% in the Tables.

Normalized Area under the Curve: We propose a novel metric which is relevant to applications where high recall is desired at very low FPR. We derive it from the ROC, as follows: we report the normalized Area under the Curve (nAUC) in a very low FPR range of FPR $= [0, ..., 1\%]$. We denote this metric by nAUCj in our results.

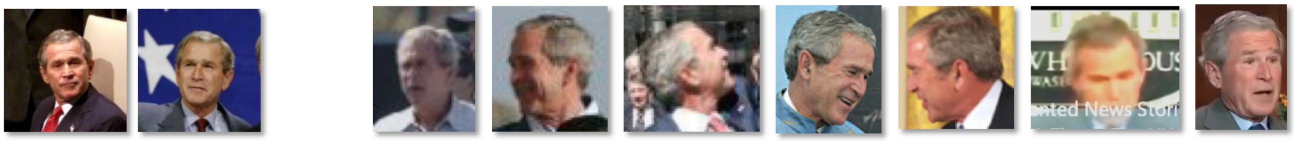
Face identification: If the protocol allows for face identification, we also report metrics designed to assess how well each method retrieves probe subjects across a pre-defined gallery. A standard tool to measure this is the Cumulative Matching Characteristic (CMC): for IJB-A identification and JANUS CS2 we also provide the recognition rate at different ranks (Rank-1, Rank-5 and Rank-10).

Template size: Unique to this work is its emphasis on reducing the number of images used to represent a template. As such, for each matching method we additionally report the number of pooled images used in practice (i.e., the number of populated bins in our representation, Sec. 4). We provide the average ± standard deviation (SD) for probe and gallery templates for each of the methods used. Importantly, besides reducing storage requirements, a smaller number of images results in fewer images being compared and hence faster template to template comparisons.

5.2. Comparison with template pooling baselines

We begin by examining the following alternative means for pooling and their effect on performance:

- **All images.** No pooling, all template images are used directly. To this end we use a CNN that was only fine



(a) LFW

(b) IJB-A

Figure 3. Qualitative comparison between facial imagery of a subject present in both LFW and IJB-A: images in LFW has a strong bias towards media collected from the web whereas the quality of IJB-A images is far more variable. Moreover LFW benchmark considers only image-pair comparisons for face verification; while IJB-A subjects are described using image templates (sets).

IJB-A identification (closed-set)										
	TPR-1%F	TPR-0.1%F	TPR-0.01%F	nAUCj	FPR-85%T	Rank-1	Rank-5	Rank-10	avg-imgG	avg-imgP
All images	85.9	71.6	51.3	63.8	0.7	82.8	92.1	94.3	24.3±20.8	7.2±13.1
Single feature pooling	83.5	68.9	50.7	62.0	1.1	83.0	91.8	94.0	1±0	1±0
Single image pooling	61.9	38.4	19.9	44.3	7.8	59.2	79.4	86.0		
Random Selection per bin	85.0	70.4	52.1	63.0	0.9	81.9	91.6	93.9	8.1±3.9	3.0±3.3
Pooled features per bin	86.1	72.3	**54.1**	63.8	0.7	82.8	91.7	93.9		
Pooled images per bin, wo/ft	86.5	72.5	53.2	64.2	0.6	83.2	91.9	94.2		
Pooled images per bin, w/ft	**87.5**	**73.5**	53.8	**65.0**	**0.5**	**84.6**	**93.3**	**95.1**		

Table 2. Comparative analysis of our proposed feature pooling per bin with other baseline methods on the IJB-A identification.

tuned once on CASIA (was not fine tuned again using pooled images).

- **Single image pooling** Pooling all images after rendering to half-profile view into a single-averaged image per template (i.e., no image binning).
- **Single feature pooling** Average pooling all the deep features extracted from all the images in a template into one feature vector. This follows similar techniques recently shown to be successful by, e.g., [7, 38]. Features were extracted using a CNN that was not trained on pooled images.
- **Random Selection per bin** Rather than pooling the images inside a bin, we randomly select one of the images as the representative for that bin. A template therefore has the same number of images used to represent it as our own method. Here too, we used a CNN that was not fine tuned on pooled images to extract the features.
- **Pooled features per bin** Same as single feature pooling above, but pooling features of each bin separately.
- **Pooled images per bin (proposed)** The method described in Sec. 4 and 4.3. Note that for this particular approach we additionally tested the effect of fine tuning the network for each of the ten training splits in each benchmark (denoted by "w/ft").

5.2.1 Analysis of Performance

All the methods presented in this paper solve the problem of comparing two sets of images and produce a score or a distance. More formally, they estimate the similarity $s(\mathcal{P}, \mathcal{G})$, given a template in the probe $\mathcal{P} = \{\mathbf{x}_1, ..., \mathbf{x}_P\}$ and another template in the gallery $\mathcal{G} = \{\mathbf{x}_1, ..., \mathbf{x}_G\}$, where $\mathbf{x}_i$ represents a feature extracted from the CNN given an image $\mathbf{I}_i$. G and P represent the cardinality of each set. Note that if both $G = 1$ and $P = 1$, then the problem is reduced to the more standard LFW matching problem. What we analyze in our work is how to exploit multiple images present in a template in the framework of the Janus benchmarks. We emphasize that the Janus benchmarks do not specify fixed sizes for both G and P, so the number of images per template varies for each testing pair, and can be as low as one.

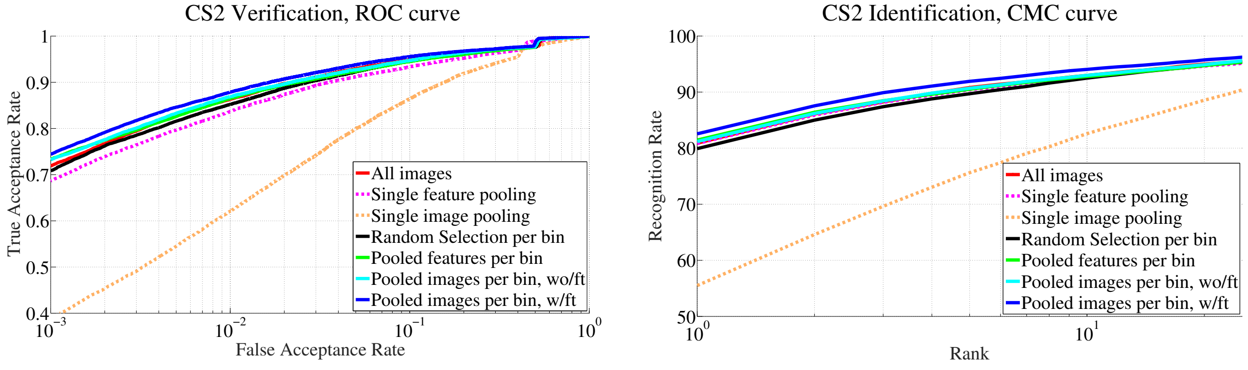
We can interpret the pooling method as a set operator which merges together features or images, changing the cardinality of the template. For instance, considering a template in the gallery, after pooling the template, $\mathcal{G}' = \text{pool}(\mathcal{G})$, its cardinality changes: $G \rightarrow G'$. As previously mentioned, this may reduce the space and time complexity required for face recognition, but it could also degrade performance if this process is not carefully performed, due to potential loss of important information.

The first baseline we examine uses no pooling, instead using all the images available in a template: this approach has complexity $\mathcal{O}(G \cdot P)$ and provides a good baseline in the Janus benchmarks, evident in Tables 2, 3 and 4.

At the opposite end are methods which pool all images together into a single, compact representation (i.e., transforming template cardinality to $G' = 1$. This is a tremendous compression in terms of media used for recognition. One interesting observation from our analysis is that pooling all features together is much more robust than pooling all the images. This can be explained by noting that deep features are trained explicitly to recognize faces and are very discriminative. Thus averaging them does not impair matching. Moreover, such methods can evaluate a template

JANUS CS2										
	TPR-1%F	TPR-0.1%F	TPR-0.01%F	nAUCj	FPR-85%T	Rank-1	Rank-5	Rank-10	avg-imgG	avg-imgP
All images	86.4	71.9	51.0	67.6	0.6	80.9	90.8	93.0	24.3±20.5	7.3±13.4
Single feature pooling	82.9	68.1	50.9	64.8	1.3	79.8	89.8	91.6	1±0	1±0
Single image pooling	62.1	38.9	20.5	46.2	7.7	55.5	75.6	82.6		
Random Selection per bin	85.2	70.8	52.6	66.8	0.8	79.9	89.7	92.5	8.2±3.9	3.0±3.3
Pooled features per bin	86.5	73.4	54.0	67.8	0.6	81.4	90.5	92.7		
Pooled images per bin, wo/ft	86.9	73.2	54.2	68.1	0.6	81.2	90.7	93.0		
Pooled images per bin, w/ft	**87.8**	**74.5**	**54.5**	**69.0**	**0.5**	**82.6**	**91.8**	**94.0**		

Table 3. Comparative analysis of our proposed feature pooling per bin with other baseline methods on the JANUS CS2 splits.


CS2 Verification, ROC curve

CS2 Identification, CMC curve

(a) (b)

Figure 4. (a) ROC and (b) CMC curves for Janus CS2 dataset considering all the tested pooling techniques.

pair at constant time as each template is represented by a single feature, yielding a complexity of $\mathcal{O}(1)$[1].

The trade-off between the two approaches is defined by compressing the media into a certain number which is more than one but still lower than the number of images in the original template, thus requiring $1 < G' \ll G$. In our case G' corresponds to the number of populated bins, and it is always far smaller than G.

Importantly, our tests show that binning and quantization play remarkable roles in partitioning the template image spaces: even naive random selection of an image within a bin improves over single feature and image pooling techniques, nearly matching the baseline performances obtained by using all template images. The complexity for these methods is equivalent to performing pair-wise bin matching, thus yielding $\mathcal{O}(G' \cdot P')$.

Performance curves for all tested methods are provided in Fig. 4 for JANUS CS2 and Fig. 5 for IJB-A. In particular in both figures we can see that the first order pooling methods, such as single feature/image pooling, represented by dashed curves in the graphs, are less robust compared to our proposed approach. Moreover, a big gap is clearly evident for single image pooling in the figures.

5.3. Comparison with state-of-the-art methods

Table 5 compares our performance to a number of existing methods on the Janus benchmarks. Our approach outperforms open-source and closed-source systems and is comparable to other published results.

In particular we largely outperform the two baselines reported in the IJB-A dataset in [26], obtained using Commercial and Government Off-The-Shelf systems ("C" and "G" in Table 5). Moreover, we consistently improve over the Open Source Biometric tool of [27] (ver. 0.5).

Our CNN based method improves over frontalized images encoded using the Fisher Vectors of [8] in all metrics. It further provides a better ROC for face verification in IJB-A than the recent method of [38]. The latter, however, achieves better results on the search protocol (identification). Interestingly, our method and [38] work in very different ways and provide different outcomes: we leverage template image pooling and a deeper network while [38] use a shallower CNN but learn a triplet similarity embedding (TSE) for each IJB-A split. This technique appears to have a better impact on identification compared to verification. Moreover, it is worth noting that [38] performs what we call "single feature pooling", which in our tests underperformed compared to the other methods we tested.

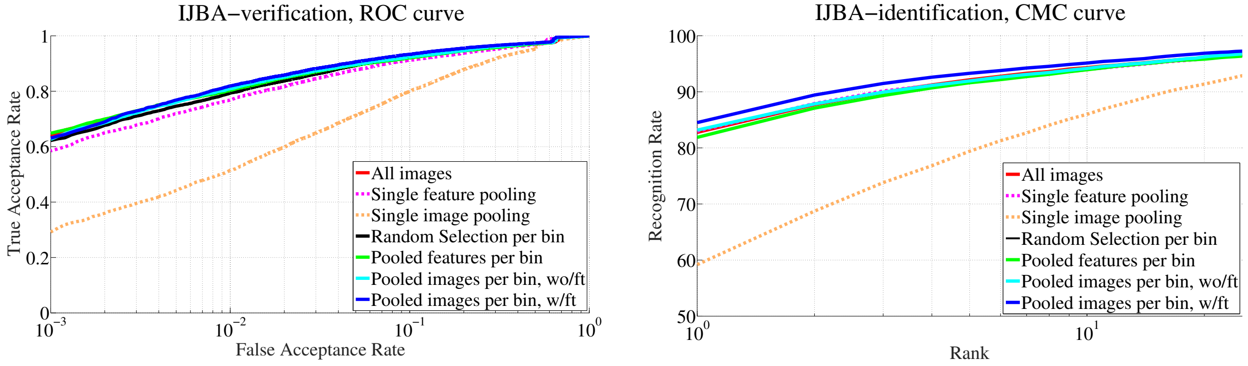
6. Conclusions

The effort to improve face recognition performance has resulted in increasingly more complex recognition pipelines. Though this process is continuously pushing performances up, the computational costs of some of these methods may not be entirely necessary. This paper turns to some of the most well established principles in image processing and computer vision – image averaging for reduced storage and computation, and improved image quality – seeking a simpler approach to representing sets of face im-

[1] Complexities do not reflect the time spent for performing pooling.

IJB-A verification							
	TPR-1%F	TPR-0.1%F	TPR-0.01%F	nAUCj	FPR-85T%	avg-imgT$_1$	avg-imgT$_2$
All images	80.8	64.0	33.3	42.6	1.7	24.3±20.8	7.3±13.4
Single feature pooling	76.8	58.5	**41.0**	40.6	2.8	1±0	1±0
Single image pooling	51.5	29.2	13.7	27.4	14.2		
Random Selection per bin	79.3	62.2	33.7	41.8	2.1	8.2±3.9	3.0±3.3
Pooled features per bin	80.3	**64.8**	27.4	42.4	1.8		
Pooled images per bin, wo/ft	81.0	62.6	35.2	42.7	1.7		
Pooled images per bin, w/ft	**81.9**	63.1	30.9	**43.1**	**1.4**		

Table 4. Comparative analysis of our proposed feature pooling per bin with other baseline methods on the IJB-A verification.



(a)

(b)

Figure 5. (a) ROC and (b) CMC curves for IJB-A benchmark considering all the tested pooling techniques.

	C[26]	G[26]	[27]	[8]	[40]	[38]	Ours
JANUS CS2							
TPR-1%F	.581	.467	–	.411	–	–	**.878**
TPR-0.1%F	.37	.25	–	–	–	–	**.745**
R. Rank-1	.551	.413	–	.381	–	–	**.826**
R. Rank-5	.694	.571	–	.559	–	–	**.918**
R. Rank-10	.741	.624	–	.637	–	–	**.940**
IJB-A Verification							
TPR-1%F	–	.406	.236	–	.732	.79	**.819**
TPR-0.1%F	–	.198	.104	–	–	.59	**.631**
IJB-A Identification							
R. Rank-1	–	.443	.246	–	.820	**.88**	.846
R. Rank-5	–	.595	.595	–	.929	**.95**	.933
R. Rank-10	–	–	–	–	–	–	**.951**

Table 5. Comparison with the state-of-the-art methods. Results that are not available are marked with a dash.

ages. We show that by aligning faces in 3D and partitioning them according to facial and imaging properties, average pooling provides a surprisingly effective yet computationally efficient approach to representing and matching face sets. Our system was tested on the most challenging benchmarks available today, the IJB-A and Janus CS2, demonstrating that not only does pooling compress template sizes and reduces the numbers of cross template comparisons it also lifts performances by noticeable margins.

Our results suggest several compelling future directions. We partition faces into expert tailored bins, determined empirically to provide optimal performances. A natural question arising from this is: can these bins be optimally determined automatically? Alternatively, pooling images (rather than features extracted from them) offer opportunities for more elaborate pooling schemes. Specifically, we pooled images using a simple, non-weighted average. A potential modification would be to explore weighted averages on image *pixels*; that is, weighing different facial regions differently according to the information they provide. Doing so, we hope to exploit facial information from multiple, partially corrupt images.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014-14071600011. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

References

[1] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Leksut, J. Kim, P. Natarajan, R. Nevatia, and G. Medioni. Face recognition using deep multi-pose

representations. In *Winter Conf. on App. of Comput. Vision*, 2016.

[2] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Continuous conditional neural fields for structured regression. In *European Conf. Comput. Vision*, pages 593–608. Springer, 2014.

[3] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search with applications to pattern recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2007.

[4] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. *Trans. Pattern Anal. Mach. Intell.*, 33(2):266–278, 2011.

[5] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2567–2573. IEEE, 2010.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[7] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *Winter Conf. on App. of Comput. Vision*, 2016.

[8] J.-C. Chen, S. Sankaranarayanan, V. M. Patel, and R. Chellappa. Unconstrained face verification using fisher vectors computed from frontalized faces. In *Int. Conf. on Biometrics: Theory, Applications and Systems*, 2015.

[9] S. Chen, C. Sanderson, M. Harandi, and B. Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 452–459, 2013.

[10] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5097–5106, 2015.

[11] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Int. Conf. Mach. Learning*, pages 376–383. ACM, 2008.

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] T. Hassner. Viewing real-world faces in 3D. In *Proc. Int. Conf. Comput. Vision*, pages 3607–3614, 2013.

[14] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.

[15] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 121–128. IEEE, 2011.

[16] G. B. Huang and E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[17] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[18] Z. Huang, R. Wang, S. Shan, and X. Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 140–149, 2015.

[19] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *Int. Conf. Mach. Learning*, pages 720–729, 2015.

[20] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. Int. Conf. Comput. Vision*, pages 605–611. IEEE, 1995.

[21] M. Irani and S. Peleg. Improving resolution by image registration. *Graphical models and image processing*, 53(3):231–239, 1991.

[22] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[23] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3304–3311. IEEE, 2010.

[24] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *Trans. Pattern Anal. Mach. Intell.*, 29(6):1005–1018, 2007.

[25] D. E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learning Research*, 10:1755–1758, 2009.

[26] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1931–1939, 2015.

[27] J. Klontz, B. Klare, S. Klum, E. Taborsky, M. Burge, and A. K. Jain. Open source biometric recognition. In *Int. Conf. on Biometrics: Theory, Applications and Systems*, 2013.

[28] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. Conf. Comput. Vision Pattern Recognition*, volume 2, pages 2169–2178. IEEE, 2006.

[29] L. Liu, B. Liu, H. Huang, and A. C. Bovik. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, 29(8):856–863, 2014.

[30] L. Liu, B. Liu, H. Huang, and A. C. Bovik. SSEQ software release. Available: live.ece.utexas.edu/research/quality/SSEQ_release.zip, 2014.

[31] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[32] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou. Multi-manifold deep metric learning for image set classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1137–1145, 2015.

[33] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *Proc. Int. Conf. Comput. Vision*, pages 329–336, 2013.

[34] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-Aware Face Recognition in the Wild. In *CVPR*, 2016.

[35] I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? *arXiv preprint arXiv:1603.07057*, 2016.

[36] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1–8. IEEE, 2007.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 2014.

[38] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *arxiv preprint*, arXiv:1602.03418, 2016.

[39] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. Int. Conf. Comput. Vision*, pages 945–953, 2015.

[40] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint*, arXiv:1507.07242, 2015.

[41] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2496–2503. IEEE, 2012.

[42] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2008.

[43] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2011.

[44] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. Available: http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html.

[45] P. Zhu, L. Zhang, W. Zuo, and D. Zhang. From point to set: Extend the learning of distance metrics. In *Proc. Int. Conf. Comput. Vision*, pages 2664–2671, 2013.