One-to-many face recognition with bilinear CNNs

Aruni RoyChowdhury   Tsung-Yu Lin   Subhransu Maji   Erik Learned-Miller

University of Massachusetts, Amherst

{arunirc,tsungyulin,smaji,elm}@cs.umass.edu

Abstract

The recent explosive growth in convolutional neural network (CNN) research has produced a variety of new architectures for deep learning. One intriguing new architecture is the bilinear CNN (B-CNN), which has shown dramatic performance gains on certain fine-grained recognition problems [15]. We apply this new CNN to the challenging new face recognition benchmark, the IARPA Janus Benchmark A (IJB-A) [12]. It features faces from a large number of identities in challenging real-world conditions. Because the face images were not identified automatically using a computerized face detection system, it does not have the bias inherent in such a database. We demonstrate the performance of the B-CNN model beginning from an AlexNet-style network pre-trained on ImageNet. We then show results for fine-tuning using a moderate-sized and public external database, FaceScrub [17]. We also present results with additional fine-tuning on the limited training data provided by the protocol. In each case, the fine-tuned bilinear model shows substantial improvements over the standard CNN. Finally, we demonstrate how a standard CNN pre-trained on a large face database, the recently released VGG-Face model [20], can be converted into a B-CNN without any additional feature training. This B-CNN improves upon the CNN performance on the IJB-A benchmark, achieving 89.5% rank-1 recall.

1. Introduction

Since the introduction of the Labeled Faces in the Wild (LFW) database [9], there has been intense interest in the problem of unconstrained face *verification*. In this problem, the goal is to determine whether two face images represent the same individual or not. From initial recognition rates in the low seventies [18], recent algorithms are achieving rates of over 99% using a variety of different methods [24, 25, 28].

While face verification has been an interesting research problem, a protocol more closely aligned with real-world applications is face *identification*, also known as *one-to-many* or *1:N* face recognition [7]. In this scenario, visual information is gathered about a set of known subjects (the *gallery*), and at test time, a new subject (the *probe*) is presented. The goal is to determine which member of the gallery, if any, is represented by the probe. When the tester guarantees that the probe is among the gallery identities, this is known as *closed set identification*. When the probes may include those who are not in the gallery, the problem is referred to as *open set identification*.

Since verification accuracy rates on LFW are rapidly approaching 100%, there has been strong demand for a new identification protocol that can push face recognition forward again. The new **IARPA Janus Benchmark A (IJB-A)** is designed to satisfy this need. The IJB-A is presented in a CVPR paper that describes the database, gives detailed information about proper protocols for use, and establishes initial baseline results for the defined protocols [12].

In this paper, we present results for IJB-A using the bilinear convolutional neural network (B-CNN) of Lin et al. [15] after adapting it to our needs and making some minor technical changes. In order to make use of images from multiple perspectives, we also investigate a technique suggested by Su et al. [23] that pools images at the feature level, rather than pooling classification scores. We follow the open-set 1:N protocol and report both cumulative match characteristic (CMC) and decision error trade-off (DET) curves, following the best practice described in the 2014 Face Recognition Vendor Test [7] and suggested in the IJB-A paper [12].

We report results on a baseline network, a network fine-tuned with a publicly available face database (FaceScrub) and also a network further fine-tuned using the IJB-A train set. Since IJB-A contains multiple images per probe, we explore two pooling strategies as well. We show that for the fine-tuned networks, and for both pooling strategies, the B-CNN architecture always outperforms the alternative, often by a large margin.

Finally, we demonstrate how a pre-trained CNN can be converted into a B-CNN without any additional fine tuning of the model. The "VGG-Face" CNN from Parkhi et al. [20]

was trained on a massive face data set which, at the time of publication, was not publicly available. However, the simplicity of the bilinear architecture allows the creation of a B-CNN from the pre-trained CNN architecture without the need for retraining the network.

In the next subsection, we detail some important properties of the IJB-A benchmark.

1.1. The IARPA Janus benchmark A

There are four key aspects of IJB-A that are relevant to this paper:

1. As stated above, it has a protocol for one-to-many classification, which is what we focus on.

2. It includes a wider range of poses than LFW, made possible in part by the fact that images were gathered by hand.

3. The data set includes both video clips and still images.

4. Each "observation" of an individual is dubbed a *template* and may include any combination of still images or video clips. In particular, a single query or probe at test time is a template and thus typically consists of multiple images of an individual.

These aspects raise interesting questions about how to design a recognition architecture for this problem.

1.2. Verification versus identification

Identification is a problem quite different from verification. In *verification*, since the two images presented at test time are required to be faces of individuals never seen before, there is no opportunity to build a model for these individuals, unless it is done at test time on the fly (an example of this is the Bayesian adaptation of the probabilistic elastic parts model [14]). Even when such on-the-fly models are built, they can only incorporate a single image or video to build a model of each subject in the test pair.

In *identification*, on the other hand, at training time the learner is given access to a gallery of subjects with which one can learn models of each individual. While in some cases the gallery may contain only a single image of a subject, in the IJB-A, there are typically multiple images and video clips of each gallery subject. Thus, there is an opportunity to learn a detailed model (either generative or discriminative) of each subject. Depending upon the application scenario, it may be interesting to consider identification systems that perform statistical learning on the gallery, and also those that do not. In this work, adaptation to the gallery is a critical aspect of our performance.

1.3. Pose variability and multiple images

An interesting aspect of IJB-A is the significant pose variability. Because the images were selected by hand, the database does not rely on a fully automatic procedure for mining images, such as the images of LFW, which were selected by taking the true positives of the Viola-Jones detector.

Handling pose One way to address pose is to build 3D models [11]. In principle, a set of gallery images can be used to generate a 3D physical model of a head containing all of the appearance information about the gallery identity.

Another approach to dealing with large pose variability is to re-render all faces from a canonical point of view, typically from a frontal pose. While this *frontalization* can be done using the above method of estimating a 3D model first, it can also be done directly, without such 3D model estimation, as done by [25]. This strategy has proven effective for obtaining high performance on face verification benchmarks like LFW [9]. It is still not clear whether such approaches can be extended to deal with the significantly higher variation in head poses and other effects like occlusion that appear in IJB-A.

Averaging descriptors. Building a generic descriptor for each image, and simply averaging these descriptors across multiple images in a set (i.e. a "template") is found to be surprisingly effective [19], yielding excellent performance in the case of verification on LFW. However, there are clear vulnerabilities to this approach. In particular, if a subject template contains large numbers of images from a particular point of view, as is common in video, such averaging may "overcount" the data from one point of view and fail to incorporate useful information from another point of view due to its low frequency in a template.

Another intriguing possibility is to build a representation which, given a particular feature representation, selects the "best" of each feature across all the images in a template. This is an approach used by the multi-view CNN of Su et al. [23] and applied to the recognition of 3D objects, given multiple views of the object. We adopt some elements of this approach and experiment with *max-pooling over feature descriptors* to get a single representation from a collection of images.

1.4. Face identification as fine-grained classification

In assessing architectures for face identification, it seems natural to consider the subarea of *fine-grained classification*. Fine-grained classification problems are characterized by small or subtle differences among classes, and often large appearance variability within classes. Popular fine-grained benchmarks include such data sets as the CUB data

set for bird species identification [27], in which each species is considered a different class. Some pairs of species, such as different types of gulls for example, have many features in common and can only be discriminated by subtle differences such as the appearance of their beaks.

Face recognition can also be viewed as a fine-grained classification problem, in which each individual represents a different class. Within-class variance is large and between-class variance is often subtle, making face recognition a natural member of the fine-grained recognition class of problems.

Recently, Lin et al. [15] have developed the *bilinear CNN (B-CNN)* model specifically for addressing fine-grained classification problems. It thus seems a natural fit for the one-to-many face recognition (identification) problem. By applying the B-CNN model to the public IJB-A data set, starting from standard network structures, we are able to substantially exceed the benchmark performance reported for the face identification task.

In the next section, we give a brief introduction to convolutional neural nets. Then in Section 3, we describe the bilinear CNN of Lin et al. [15]. In Section 4, we discuss experiments and results, and we end in Section 5 with some conclusions and future directions.

2. Introduction to CNNs

Convolutional neural networks (CNNs) are composed of a hierarchy of units containing a convolutional, pooling (e.g. max or sum) and non-linear layer (e.g. ReLU $\max(0, x)$). In recent years deep CNNs typically consisting of the order of 10 or so such units and trained on massive labelled datasets such as ImageNet have yielded generic features that are applicable in a number of recognition tasks ranging from image classification [13], object detection [6], semantic segmentation [8] to texture recognition [3].

In the domain of fine-grained recognition, such as identifying the breed of a dog, species of a bird, or the model of a car, these architectures, when combined with detectors that localize various parts of the object, have also yielded state-of-the-art results. Without the part localization, CNNs typically don't perform as well since there is a tremendous variation in appearance due to different poses that instances of a category can be in. This pose variation overwhelms the subtle differences across categories, a phenomenon typical also in the face recognition problem. However, the drawback of these approaches is that they require (a) manual annotation of parts which can be time-consuming, (b) the detection of parts which can be computationally expensive.

In contrast, models originally developed for texture recognition such as Bag-of-Visual Words (BoVM) [4] and their variants such as the Fisher vector [21] or VLAD [10], have also demonstrated good results on fine-grained recognition tasks. These models don't have an explicit model-

a.

b.

c.

d.

Figure 1. Filters learned from the bilinear CNN. Each 2x8 group of image patches shows the top-K patches in the input images that gave the highest response for a particular 'conv5+relu' filter of a symmetric B-CNN, using VGG 'M' networks and trained on the FaceScrub dataset [17]. **a.** The first 4 sets of filter responses show traditional facial features such as eyes, eyes+eyebrows, partially open mouth, and noses. **b.** The B-CNN also learns categories of features that seem to be related to accessories, such as eyeglasses and jewelry. **c.** The features in this row seem to be correlated with hair (both facial hair and hair on the top of the head). **d.** Finally, these features are associated with possibly informative aspects of the background such as text and spurious lines.

ing of parts, nor do they require any annotations, making them easily applicable to new domains. Deep variants of Fisher vectors [3] based on features extracted from the convolutional layers of a CNN trained on ImageNet [5] provide a better alternative to those based on hand-crafted features such as SIFT [16]. However, pose normalization such as "frontalization" for faces, or part detection for birds, followed by a CNN trained for fine-grained recognition outperforms these texture models. See for example DeepFace of Facebook [25], or pose-normalized CNNs for birds species identification [1, 30].

3. Bilinear CNNs

The bilinear CNN model, originally introduced by Lin et al. [15], bridges the gap between the texture models and part-based CNN models. It consists of two CNNs whose convolutional-layer outputs are multiplied (using outer product) at each location of the image. The resulting bilinear feature is pooled across the image resulting in

an orderless descriptor for the entire image. This vector can be normalized to provide additional invariances. In our experiments we follow the same protocol as [15] and perform signed square-root normalization ($\mathbf{y} \leftarrow \text{sign}(\mathbf{x})\sqrt{|\mathbf{x}|}$) and then ℓ_2 normalization ($\mathbf{z} \leftarrow \mathbf{y}/\|\mathbf{y}\|$).

If one of the feature extractors was a part detector and the other computed local features, the resulting bilinear vector can model the representations of a part-based model. On the other hand, the bilinear vector also resembles the computations of a Fisher vector, where the local features are combined with the soft membership to a set of cluster centers using an outer product (to a rough approximation, see [15] for details).

A key advantage is that the bilinear CNN model can be trained using *only* image labels without requiring ground-truth part-annotations. Since the resulting architecture is a directed acyclic graph (DAG), both the networks can be trained simultaneously by back-propagating the gradients of a task-specific loss function. This allows us to initialize generic networks on ImageNet and then fine-tune them on face images. Instead of having to train a CNN for face recognition from scratch, which would require both a search for an optimal architecture and a massive annotated database, we can use pre-trained networks and adapt them to the task of face recognition.

When using the *symmetric B-CNN* (both the networks are identical), we can think of the bilinear layer being similar to the quadratic polynomial kernel often used with Support Vector Machines (SVMs). However, unlike a polynomial-kernel SVM, this bilinear feature is pooled over all locations in the image and can be trained end-to-end.

4. Experiments

In our experiments section, we first describe the various protocols and datasets used in training our models and evaluating our approach. In particular, the open-set protocol for face identification and the various metrics used in the IJB-A benchmark are explained briefly. Next, we describe the various experimental settings for our methods. This includes details on data pre-processing, network architectures, fine-tuning procedure and using pre-trained models.

4.1. Datasets and protocols

The **IJB-A** face recognition protocol [12] provides three sets of data for each of its 10 splits. Models can be learned on the *train set*, which contains 333 persons with varying number of images, including video frames, per person. The *gallery set* consists of 112 persons. The *probe set* is comprised of imagery from 167 persons, 55 of whom are not present in the gallery (known as "distractors" or "impos-
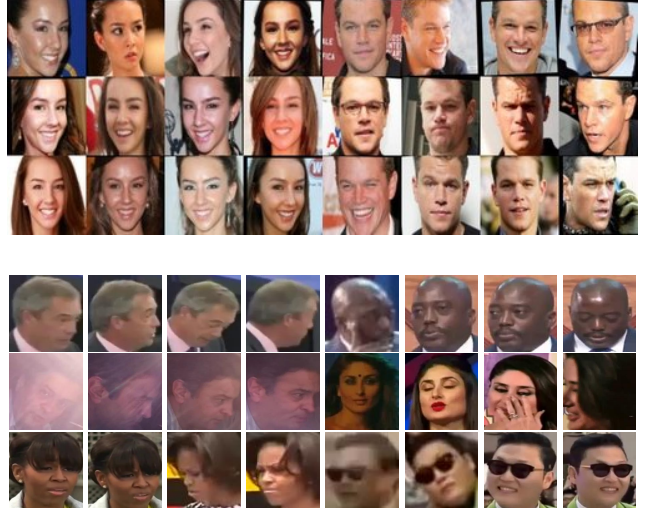

Figure 2. Sample images from the **FaceScrub**[2] *(top)* and **IJB-A** *(bottom)* datasets.

tors"). It follows the *open-set* protocol in its identification task.

The IJB-A benchmark comprises both a one-to-many recognition task (identification) and a verification task. We focus in this paper on the identification task. The details of the identification protocol and reporting of results can be found in the NIST report by Grother et al. [7]. To evaluate the performance of a system in correctly matching a probe template to its identity (from among the identities present in the gallery set), the *Cumulative Match Characteristic (CMC)* curve is used. This summarizes the accuracy on probe templates that have a match among the gallery identities at various ranks. The rank-1 and rank-5 values are individual points on this curve, which usually reports recall from ranks 1 to 100.

In the open-set protocol, two particular scenarios may arise as follows: firstly, the "non-match" or "impostor" templates might be wrongly classified as a gallery identity, if the classifier score from the one-versus-rest SVM for that identity is above some threshold (false alarms). Secondly, a template that is genuinely from among the gallery identities may be wrongly rejected if all the SVM scores for it are below some threshold (misses). The *Decision Error Trade-off (DET)* curve plots false alarm rate or false positive identification rate (FPIR) and miss rate or false negative identification rate (FNIR) by varying this threshold, capturing both of the scenarios mentioned above. As specified in the IJB-A benchmark, we report FNIR at FPIR's of 0.1 and 0.01.

The **FaceScrub** dataset [17] is an open-access database of face images of actors and actresses on the web, provided as hyperlinks from where the actual images can be downloaded. It contains 530 persons with 107,818 still images in total. There are on average 203 images per person. In

an additional experiment, we use this external data to first fine-tune the networks, before subsequent fine-tuning on the IJB-A train data. All overlapping identities between the two datasets are removed from FaceScrub before training the networks on it. As some of the download links provided in FaceScrub were broken (and after overlap removal) we finally train the networks on 513 identities, having a total of 89,045 images. We keep a third of the images in each class as validation sets and use the rest for training the networks.

4.2. Methods

Pre-processing

The bounding boxes provided in the IJB-A metadata were used to crop out faces from the images. The images are resized according to the normalization parameters specified in the architecture details of a particular network (see below). This resizing does not maintain the aspect ratio.

Network architectures

As a baseline for deep models, we use the Imagenet-pretrained "M-net" model from VGG's MatConvNet [26]. All results using this network architecture are hereafter referred to as **"CNN"**. We consider the network outputs of the fully-connected layer after rectification, i.e. layer-19 ('fc7' + 'relu7') to be used as the face descriptor. An input image is resized to 224×224 following the way the network had been initially trained on Imagenet, resulting in a 4096-dimensional feature vector.

We use a symmetric bilinear-CNN model, denoted from now on as **"B-CNN"**, that has both Network A and Network B set to the "M-net" model. Similar to the procedure followed in [15], the bilinear combination is done by taking the rectified outputs of the last convolutional layer in each network, i.e. layer-14 ('conv5' + 'relu5'). We chop off both the networks at layer 14, add the bilinear combination layer, a layer each for square-root and L2 normalization, and then a softmax layer for classification. For this architecture, the image is upsampled to be 448×448, resulting in a $27 \times 27 \times 512$ output from each network at layer-14 (27×27 are the spatial dimensions of the response map and 512 denotes the number of CNN filters at that layer). The bilinear combination results in a 512×512 output, and its vectorization (followed by the normalization layers mentioned earlier) gives us the final face descriptor.

Network fine-tuning

The models described in this set of experiments were trained initially for large-scale image classification on the Imagenet dataset. Fine-tuning the networks for the specific task of face recognition is expected to significantly boost performance. We consider three different scenarios with respect to fine-tuning:

- *no-ft:* No fine-tuning is done. We simply use the Imagenet-pretrained model without any retraining on face images as a baseline for our experiments.

- *FaceScrub:* The Imagenet-pretrained network is fine-tuned on the FaceScrub dataset by replacing the last layer with a softmax regression and running back-propagation with dropout regularization of 0.5 for 30 epochs. We begin fine-tuning with a learning rate of 0.001 for the lower layers and 0.01 for the last layer and divide them both by 10 if the validation error rate does not change. The stopping time is determined when the validation error remains constant even after learning rate is changed. The B-CNN is similarly fine-tuned for 70 epochs on FaceScrub data.

- *FaceScrub+Train:* The FaceScrub data provides a good initialization for the face identification task to the networks, following which we fine-tune on the IJB-A train set for 30 epochs in case of the regular CNN and 50 epochs for the B-CNN. The fine-tuning on FaceScrub gives us a single model each for CNN and B-CNN. In the current setting, we take this network and further fine-tune it on each of the train sets provided in the 10 splits of IJB-A. This setting considers fine-tuning the network on images that are closer in appearance to the images it will be tested upon, i.e., the train set of IJB-A, being drawn from the same pool of imagery as the probe and gallery sets, is more similar to the test images than FaceScrub images.

Classifiers and pooling

One-versus-rest linear SVM classifiers are trained on the gallery set for all our experiments. We do not do any form of template-pooling at this stage and simply consider each image or video frame of a person as an individual sample. The weight vectors of the classifiers are rescaled such that the median scores of positive and negative samples are +1 and -1. Since all evaluations on this protocol are to be reported at the template level, we have the following straightforward strategies for pooling the media (images and video-frames) within a template at test time:

- *Score pooling:* We use the max operation to pool the SVM scores of all the media within a probe template. This is done after SVM scores are computed for each individual image or frame that comprises a template.

- *Feature pooling:* The max operator is applied on the features this time to get a single feature vector for a template. Thus, the SVM is run only once per probe template.

Conversion of a pre-trained CNN to a B-CNN

In addition to the Imagenet-pretrained networks described above, we also ran experiments in modifying a standard CNN, trained on a large face database, to convert it into a B-CNN. We use the VGG-Face model from Parkhi et al. [20][3] where they trained the `imagenet-vgg-verydeep-16` network [22] to do face identification using a large data set of faces which contained no overlap in person identities with IJB-A. This deep network architecture, pre-trained on such a large database of 2.6 million images, produced very high results on IJB-A (see results section).

We replicated this network to form a symmetric B-CNN, as described in Section 4.2, and for each of the 10 splits in IJB-A, trained SVMs on the output features for each gallery identity. Note that this "gallery training" does not change the parameters of the original core CNN network or the B-CNN produced from it.

5. Results

We report two main sets of results. The first set of experiments shows that B-CNNs consistently outperform standard CNNs when fine-tuned on generic face data and also when adapted further to the specific data set at hand. The second set of results, using the pre-trained VGG-Face network, shows that even in a scenario when such massive training data is not readily available to the end user, a B-CNN can be constructed, without further network training, that improves the final accuracy of the network.

For both sets of results, we evaluate the task of *open-set* identification on this dataset, also referred to as the "1:N identification task" in the IJB-A benchmark report [12]. We include the *cumulative match characteristic (CMC)* curve plots which summarizes the accuracy of the model over a range of ranks from 1 to 100.

5.1. Fine-tuned networks

Comparison of fine-tuning methods

The various fine-tuning settings that are used here and the performance of the models at *rank-1* and *rank-5* are shown in the rows of Table 1. Performance across ranks is shown in the CMC curves in Figure 3.

Without any fine-tuning, the Imagenet-pretrained networks give low accuracy in both models — 28.9% for the standard CNN and 31.2% for the bilinear CNN.

Fine-tuning on FaceScrub makes the networks specific to the task of identifying faces, and we can see highly specialized filters being learned in the convolutional layer of the network in Figure 1. Consequently, the performance

[3]Downloadable from their site `http://www.robots.ox.ac.uk/~vgg/software/vgg_face/`

of both CNNs and B-CNNs improves, as seen in Figure 3. However, the B-CNN model fine-tuned on FaceScrub outperforms the CNN by almost 10% (52.5% versus 44.5%).

Rank-1	CNN		B-CNN	
	score pooling	feature pooling	score pooling	feature pooling
no-ft	0.289 ± 0.028	0.281 ± 0.023	0.312 ± 0.029	0.297 ± 0.037
FaceScrub	0.445 ± 0.020	0.421 ± 0.021	0.521 ± 0.021	0.525 ± 0.019
FaceScrub+Train	0.536 ± 0.020	0.526 ± 0.019	0.579 ± 0.014	**0.588 ± 0.020**
Rank-5				
no-ft	0.519 ± 0.026	0.490 ± 0.022	0.517 ± 0.029	0.500 ± 0.022
FaceScrub	0.684 ± 0.024	0.658 ± 0.024	0.732 ± 0.017	0.729 ± 0.019
FaceScrub+Train	0.778 ± 0.018	0.761 ± 0.018	**0.797 ± 0.018**	0.796 ± 0.017

Table 1. Showing rank-1 and rank-5 retrieval rates of CNN and B-CNN on the IJB-A 1:N identification task.

The final fine-tuning setting takes the FaceScrub trained networks and further fine-tunes them on the train set of IJB-A. Unsurprisingly, using training data that is very close to the distribution of the test data improves performance across ranks, as is shown in Figure 3. The CNN performance at rank-1 improves from 44.5% to 53.6%, while the B-CNN performance rises from 52.5% to 58.8%. The IJB-A train set images are very similar to the type of images in the gallery and probe test sets. This effect is also more prominent given that the original training dataset, FaceScrub, is not very extensive (530 celebrities, 107,818 still images) compared to the large degree of variations possible in faces and may not cover the type of images (low resolution, motion blurs, video frames) seen in IJB-A (see the sample images in Figure 2). However, even without depending upon the IJB-A train set for further fine-tuning, the performance of B-CNN using only the single external FaceScrub dataset substantially improves over the reported baselines on IJB-A (52.5% versus 44.3% as shown in Table 2).

In Figure 4, we can see the consistent and large improvement in relative performance by using the bilinear CNN versus the regular CNN over the entire range of ranks. The B-CNN also outperforms the highest baseline on IJB-A, "GOTS" by a large margin.

Comparison of template-pooling methods

We also evaluate two pooling schemes in Table 1 at score and feature level using rank-1 and rank-5 retrieval rates. For the CNN network, the two pooling schemes achieve almost the same accuracy, while feature pooling either outperforms score pooling or has negligible difference when using fine-tuned B-CNN models. The reason for the preference of feature pooling on B-CNN is that the model learns good semantic part filters by fine-tuning (see Figure 1) and feature pooling combines the part detections which might be invisible from a particular viewpoint by taking the maximum part detection response across views when a sequence of faces (with different views) is provided. The impact of feature-pooling becomes less evident at higher ranks.
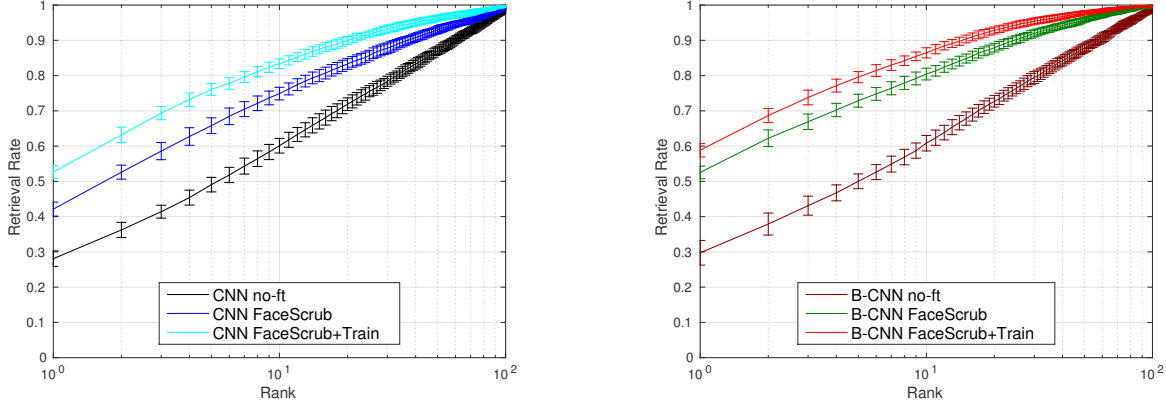
Figure 3. CMC curves showing the effect of fine-tuning on the standard CNN *(left)* and B-CNN *(right)* models, evaluated on the IJB-A dataset using feature pooling of templates. Both models show a large improvement when fine-tuned on FaceScrub when compared to the performance of Imagenet-pretrained networks without fine-tuning. Further fine-tuning on the IJB-A train set after fine-tuning on FaceScrub increases performance in both the models as the test and training domains are now more similar.
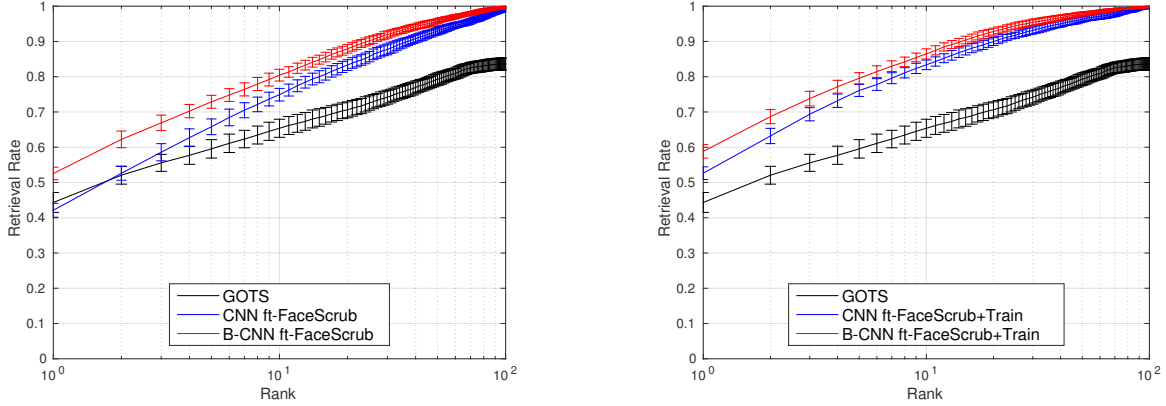

Figure 4. CMC curves comparing the standard CNN and the B-CNN models evaluated on IJB-A with feature-pooling of templates; *left:* models fine-tuned on FaceScrub; *right:* the FaceScrub models are fine-tuned further on IJB-A train set. The B-CNN does better than the CNN and also outperforms the "GOTS" baseline method [12] by a large margin.

Comparison with IJB-A benchmark

Table 2 shows performance on IJB-A with respect to the published benchmark. The best-performing B-CNN model, which is trained on both FaceScrub and subsequently on IJB-A train set data, exceeds the IJB-A baselines [12] by a large margin — 58.8% versus 44.3% at rank-1 and 79.6% versus 59.6% at rank-5, when compared against the highest performing GOTS (government off-the-shelf) baseline.

At FPIR's of 0.1 and 0.01, the FNIR of B-CNN versus GOTS is 65.9% versus 76.5% and 85.7% versus 95.3%, respectively. The DET curves for the bilinear and regular CNN models are shown in Figure 5. Our system, using learned bilinear features followed by the one-versus-rest linear SVMs trained on the gallery identities, is robust to the impostor probes without adversely affecting recognition of matching probe templates.

IJB-A	OpenBR	GOTS	B-CNN
rank-1	0.246 ± 0.011	0.443 ± 0.021	$\mathbf{0.588 \pm 0.020}$
rank-5	0.375 ± 0.008	0.595 ± 0.02	$\mathbf{0.796 \pm 0.017}$
FNIR @ FPIR=0.1	0.851 ± 0.028	0.765 ± 0.033	$\mathbf{0.659 \pm 0.032}$
FNIR @ FPIR=0.01	0.934 ± 0.017	0.953 ± 0.024	$\mathbf{0.857 \pm 0.027}$

Table 2. We compare the performance of B-CNN model with the baselines reported on **IJB-A** [12]. The highest performing bilinear model (fine-tuned on FaceScrub and IJB-A train set) is considered for this comparison.

Run-time and feature size information

The experiments were run on an NVIDIA Tesla K40 GPU. The B-CNN encoding time for a single image is roughly 0.0764 seconds. The one-versus-rest SVM training on the gallery takes around 17 seconds on average for a single per-

son, using the 262,144-dimensional B-CNN features. This procedure can be easily parallelized for each person enrolled in the gallery. The pooling methods depend largely on the number of images comprising a single template. On average the feature pooling is 2 times faster than score pooling at negligible difference in accuracy. Fine-tuning the bilinear CNN network for 70 epochs on FaceScrub took about 2 days.
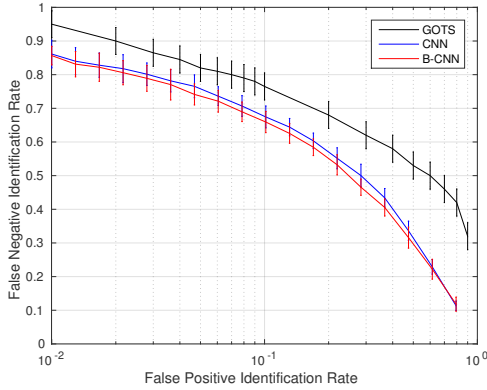

Figure 5. DET curve comparing the best-performing (FaceScrub+Train fine-tuned) standard CNN and the B-CNN models evaluated on IJB-A. The performance of B-CNN is slightly better than that of CNN (lower values mean better).

5.2. Pre-trained networks

We now demonstrate that the bilinear layer can be applied out-of-the-box to further boost the high face recognition performance of an existing powerful pre-trained network. We evaluate the VGG-Face network [20] on IJB-A as a baseline, using one-versus-rest SVM classifiers and no data augmentation at test time, consistent with the settings followed in our previous set of experiments. We compare the performance of a B-CNN built using the pre-trained VGG-Face network against the CNN baseline (the unmodified pre-trained network) and the method of Chen et al. [2], who also use a deep network trained on a large external face dataset. Table 3 summarizes the results, where we can see that even without fine-tuning the bilinear model, the B-CNN (at 89.5%) outperforms both the baseline accuracy of the pre-trained CNN (89.2%) by a small margin as well as the method of Chen et al. (86%).

Without training the bilinear model, we see a only a small improvement over the regular VGG-Face CNN, unlike the large increase in performance observed in the previous experiments when fine-tuning of the bilinear model could be done. This leads us to believe that re-training the entire bilinear network formed out of VGG-Face on a suitably large dataset is necessary to boost the performance to a significant level (this is not done in our present work).

IJB-A	Chen et al. [2]	VGG-Face [20]	B-CNN
rank-1	0.86 ± 0.023	0.892 ± 0.010	$\mathbf{0.895 \pm 0.011}$
rank-5	0.943 ± 0.017	0.960 ± 0.006	$\mathbf{0.963 \pm 0.005}$

Table 3. We compare the performance of the pre-trained VGG-Face [20] network with B-CNN, regular CNN and the deep CNN of Chen et al. [2]. Score pooling is used in both methods.

6. Discussion

It is perhaps not surprising that CNN architectures that have succeeded on other fine-grained recognition problems also do well at face identification, both after fine-tuning and also as a simple modification to pre-trained models.

There are a number of directions to continue exploring these models:

- Re-training the entire model using an objective similar to the Multi-view CNN objective [23] in which the parameters of the network are learned under the assumption that the max will be taken across the multiple images in a template.

- Using datasets much larger than FaceScrub, such as the CASIA WebFaces [29] (with 10,000 identities and half-a-million images) to train the network, should further improve the performance.

- Training a very deep architecture from scratch on a sufficiently large face dataset, instead of fine-tuning pre-trained networks.

We believe the success of the B-CNN relative to non-bilinear architectures makes them a sensible starting point for a wide variety of continued experiments.

Acknowledgements

References

[1] S. Branson, G. V. Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *Proc. BMVC*, 2014.

[2] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep CNN features. *CoRR*, abs/1508.01722, 2015.

[3] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and description. In *Proc. CVPR*, 2015.

[4] G. Csurka, C. R. Dance, L. Dan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. ECCV Workshop on Stat. Learn. in Comp. Vision*, 2004.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

[6] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[7] P. Grother and M. Ngan. Face recognition vendor test (FRVT): Performance of face identification algorithms. In *NIST Interagency report 8009*, 2014.

[8] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. ECCV*, 2014.

[9] 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, 2007.

[10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.

[11] I. Kemelmacher-Shlizerman and R. Basri. 3D face reconstruction from a single image using a single reference face shape. *PAMI*, 2011.

[12] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *Proc. CVPR*, 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

[14] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *Proc. CVPR*, 2013.

[15] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *Proc. ICCV*, 2015.

[16] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

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

[18] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. CVPR*, 2007.

[19] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proc. CVPR*, 2014.

[20] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[21] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV*, 2015.

[24] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Proc. NIPS*, 2014.

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

[26] A. Vedaldi and K. Lenc. Matconvnet-convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.

[27] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

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

[29] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014.

[30] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *Proc. CVPR*, 2012.