The Do's and Don'ts for CNN-based Face Verification

Ankan Bansal Carlos Castillo Rajeev Ranjan Rama Chellappa
UMIACS
University of Maryland, College Park
{ankan,carlos,rranjan1,rama}@umiacs.umd.edu

Abstract

While the research community appears to have developed a consensus on the methods of acquiring annotated data, design and training of CNNs, many questions still remain to be answered. In this paper, we explore the following questions that are critical to face recognition research: (i) Can we train on still images and expect the systems to work on videos? (ii) Are deeper datasets better than wider datasets? (iii) Does adding label noise lead to improvement in performance of deep networks? (iv) Is alignment needed for face recognition? We address these questions by training CNNs using CASIA-WebFace, UMDFaces, and a new video dataset and testing on YouTube-Faces, IJB-A and a disjoint portion of UMDFaces datasets. Our new data set, which will be made publicly available, has 22,075 videos and 3,735,476 human annotated frames extracted from them.

1. Introduction

The re-emergence of deep convolutional neural networks has led to large improvements in performance on several face recognition and verification datasets [12, 20, 17]. However, the face recognition problem isn't "solved" yet. The process of training a face recognition system starts with choosing a dataset of face images, detecting faces in images, cropping and aligning these faces, and then training deep networks on the cropped and possibly aligned faces. Every step of the process involves many design issues and choices.

Some issues have received significant attention from researchers. These include choices about the architecture of neural networks. On the other hand, there are several other design choices which require more attention. These arise at every stage of the process from face detection and thumbnail (image obtained after cropping and aligning the face image) generation to selecting the training dataset itself. We tackle some of these design questions in this paper. We also introduce a dataset of 22,075 videos collected from YouTube of 3,107 subjects. These subjects are mainly from batch-1 of the recently released UMDFaces [2] dataset. We release face annotations for 3,735,476 frames from these videos and the corresponding frames separately. We use this dataset to study the effect of using a mixture of video frames and still images on verification performance for unconstrained faces such as the IJB-A [17] and YTF [37] sets.

Face detection is the first step in any face recognition pipeline. Several CNN-based face and object detectors have been introduced which achieve good detection performance and speeds [26, 30, 29, 21, 27, 40, 28, 11]. Each of these detectors learns a different representation. This leads to generation of different types of bounding boxes for faces. Verification accuracy can be affected by the type of bounding box used. In addition, most recent face recognition and verification methods [35, 31, 33, 5, 10, 34] use some kind of 2D or 3D alignment procedure [41, 15, 28, 8]. All these variables can lead to changes in performance of deep networks. To the best of our knowledge there has been very little systematic study of effects of the thumbnail generation process [25] on the accuracy of deep networks. In section 3.4 we study the consequences of using different thumbnail generation methods. We show that using a good keypoint detection method and aligning faces both during training and testing leads to the best performance.

Other questions concern the dataset collection and cleaning process itself. The size of available face datasets can range from a few hundred thousand images [2, 39, 24, 37] to a few million [16, 23, 7, 6, 25]. Other datasets, which are not publicly available, can go from several million faces [35] to several hundred million faces [32]. Much of the work in face recognition research might behave differently with such large datasets. Apart from datasets geared towards training deep networks, some datasets focus on evaluation of the trained models [12, 17]. All of these datasets were collected using different methodologies and techniques. For example, [37] contains videos collected from the internet which look quite different from still image datasets like [39, 2, 7]. We study the effects of this difference between still images and frames extracted from

videos in section 3.1 using our new dataset. We found that mixing both still images and the large number of video frames during training performs better than using just still images or video frames for testing on any of the test datasets [17, 37, 2].

In section 3.2, we investigate the impact of using a deep dataset against using a wider dataset. For two datasets with the same number of images, we call one deeper than the other if on average it has more images per subject than the other. We show that the choice of the dataset depends on the kind of network being trained. Deeper networks perform well with deeper datasets and shallower networks work well with wider datasets.

Label noise is the phenomenon of assigning an incorrect label to some images. Label noise is an inherent part of the data collection process. Some authors intentionally leave in some label noise [25, 6, 7] in the dataset in hopes of making the deep networks more robust. In section 3.3 we examine the effect of this label noise on the performance of deep networks for verification trained on these datasets and demonstrate that clean datasets almost always lead to significantly better performance than noisy datasets.

We make the following main contributions in this paper:

- We conduct a large scale systematic study about the effects of making certain apparently routine decisions about the training procedure. Our experiments show that data diversity, number of individuals in the dataset, quality of the dataset, and good alignment are keys to obtaining good performance.
- We suggest some general rules that could lead to improvement in the performance of deep face recognition networks. These practices will also guide future data collection efforts.
- We introduce a large dataset of videos of over 3,000 subjects along with 3,735,476 human annotated bounding boxes in frames extracted from these videos.

2. Dataset

Still photos from the internet cannot match the amount of variation that videos provide. Videos (and frames extracted from the videos) are under-utilized because of the difficulty in cleaning and annotating the data. There is a need for effective methods for annotating video data. We describe a new dataset[1] aimed at face recognition research. It contains 22,075 videos of 3,107 subjects collected from YouTube. We provide bounding box annotations for 3,735,476 frames from the videos. We explain our methodology of collecting this dataset which, we hope, will be useful to researchers working on face verification and related problems.

[1] http://umdfaces.io/

2.1. Collecting data

We searched YouTube for over 3000 subject identities (from batch-1 of UMDFaces [2]) and tried to download the first 20 videos for each person. We used the open source system youtube-dl [1] for searching and downloading the videos. We downloaded a total of about 40,000 videos.

2.2. Automated filtering

From each video, we extracted either all the frames or the first 4,000 frames, whichever is lower. This process gave us over 140 million frames. We randomly selected about 10% of these frames to process further. Next we detected faces in the retained frames. At the time of collecting this data, most detection systems were too slow to be of use. Faster RCNN [30] claimed to detect objects at 7 frames per second. We decided to use the, then newly introduced, YOLO detector [29]. We trained the YOLO detector on the WIDER dataset [38] and fine-tuned on the FDDB dataset [13]. We were able to run the trained detector at about 25 frames per second on a Titan X GPU. This enabled us to detect all faces in these 14 million images in less that one week. This gave us over 40 million face boxes in 14 million frames. We again randomly selected 4000 face boxes for each subject identity finally leaving ourselves with about 14 million boxes.

Our next task was to remove all face box proposals which did not belong to the person in question. We used the all-in-one method proposed in [28] to detect key landmark points on each face and used them to align the faces. We used the images in batch-1 of the UMDFaces dataset [2] as reference images for the subjects. Our problem now reduced to a verification problem. For each subject we need to verify whether a face box belongs to that person.

We used the verification method proposed in [5] for filtering the proposal boxes which are not of the person in question. We extracted features (using a network trained in the same way as [39, 5]) for all images in batch-1 of UMDFaces [2] and take their average over a subject to obtain one feature vector for the subject. Then, for each face box in our dataset for the subject, we compared the feature vector with the reference feature vector obtained above and kept only those boxes with similarity with the reference feature vector above a threshold. We used cosine similarity as the similarity metric and used a low threshold to avoid removing the hard-positive examples from the dataset as these are very valuable. This leaves us with about 4 million face boxes.

2.3. Crowd-sourcing final filtering

To obtain the final dataset, we use Amazon Mechanical Turk (AMT) to filter the proposals. We show each proposal to 2 'mechanical turkers'. Each screen in our AMT task contains 50 images to be filtered and 3 reference images of a subject obtained from the UMDFaces dataset [2]. We requested the mechanical turkers to select images which

Figure 1: Some sample annotated bounding boxes from the dataset. Each row contains frames from a video. There is a large amount of pose and expression variation in each video.

do not belong to the subject under consideration. We removed all the faces boxes which were selected by at least one turker. To ensure high quality annotations, we adopted the following quality control method.

2.4. Quality control through sentinels

We used the method similar to the one used in [3, 2] for controlling the quality of annotations. Each screen of 50 images contains 5 known images of another subject. Depending on whether the turkers select these 'sentinel' images, they get an accuracy score. We only considered the votes of turkers with high accuracy scores.

2.5. Result

After the final filtering through human annotators, we have 3,735,476 annotated frames in 22,075 videos. We will publicly release this massive dataset for use by the computer vision community.

Next, we use this dataset to show the importance of using video frames for training while testing on real world scenarios like IJB-A and YTF in section 3.1. We show that utilizing the vast amounts of video data and mixing video frames with still images can give a significant boost in verification performance over using only video frames or still images.

3. Questions and Experiments

We show that judicious decisions about the training set and procedures can lead to large improvements in verification accuracy of deep networks. We first use the introduced dataset to show the importance of using video frames while training for verification. Then we investigate some more questions that will guide researchers towards good practices for training deep networks for face verification and identification. These include: (i) whether deep datasets are better than wide datasets (section 3.2); (ii) whether label noise helps in improving performance (section 3.3); and (iii) how important is the thumbnail generation method for training and testing deep networks (section 3.4). We use the Caffe [14] framework for all experiments.

3.1. Do deep recognition networks trained on stills perform well on videos?

Images in most still image datasets [12, 22, 19] are taken with high quality cameras in good lighting. Photos of celebrities on the internet are often selected from among several taken by a professional photographer. This introduces a bias towards high quality images. Models trained on only still images perform poorly on frames extracted from videos [17]. These frames are extremely challenging because of pose, expression, and lighting variations. At the same time, models trained only on videos perform poorly on still images. There is a huge amount of video data available and only a limited number of still images. We show that training on a mixture of images and video frames is really important for achieving good verification performance.

We train deep networks on the following five sets and compare the verification performance of these networks:

- **Stills**: Some part (batch-1) of the UMDFaces [2] dataset. This comprises of about 140,000 still images. We train an Alexnet-derived architecture [31] on these images for 100,000 iterations with a batch size of 128 and initial learning rate of 0.01 and reduced by half every 15,000 iterations.
- **Frames**: The same number (140,000) of video frames from our dataset (section 2). Each subject has the same number of images as the case above (Stills). We used the same training method as above.
- **Frames++**: The same number of subjects as above but using many more video frames per subject for a total of about 1 million video frames. We trained this model

for 100,000 iterations and decreased the learning rate by half every 20,000 iterations.

- **Mixture**: A mixture of still images and video frames from UMDFaces and our dataset. We took 50% of images from batch-1 of UMDFaces and the other 50% from our video frames dataset for a total of about 140,000 images. We trained this network for 100,000 iterations.
- **Mixture++**: The same number of still images as 'Stills' but about 1 million video frames. We again train the network on this dataset for 100,000 iterations.

Note that we are using far more images in the Frames++ and Mixture++ cases than the other cases. However, we believe that it is fair to compare these five methods because it is much easier to obtain millions of video frames than to obtain millions of still images. There is a lot more variation in 100 images than there is in 100 continuous video frames. Also, in real world scenarios, the amount of video data is increasing rapidly and the majority of recognition has to happen in videos.

We use an architecture [31] derived from Alexnet [18] due to it's easy availability and practicality. It is very fast to train and is perfectly suited for large-scale experiments like ours. Also, it provides excellent results [31]. However, we believe that, in this case, our observations are general and will be valid for other network architectures too.

We trained networks on these five sets and compare performance of the trained models on IJB-A [17], YouTube Faces datasets [37] and batch-3 of the UMDFaces [2] dataset. In this experiment and the rest of the paper, unless otherwise stated, we train the same architecture of networks on different datasets for a fixed number of iterations (100,000). We adopt the 1:1 verification protocol similar to the one introduced in [17] for evaluating the performance of these deep networks. We give a brief description of the evaluation protocol next.

3.1.1 Protocol

The IJB-A 1:1 verification protocol [17] uses a decision error tradeoff (DET) curve for evaluation. The DET curve is equivalent to an ROC curve. In our examples we evaluate the performance for 1:1 verification on pairs of images or templates for different datasets [12, 37, 2]. For all experiments, we use ROC curves for evaluation.

3.1.2 Results

Figures 2 and 3 show the performance of the above five experiments. They clearly show the importance of using a mixture of video frames and still images for all cases. We see that while the performance of the 'Stills' and 'Mixture' cases is very close for both IJB-A and UMDFaces, the performance of 'Frames' is very poor. This is because of the presence of many still images in the test sets and the low variety in the few training frames. On the other hand, note that the performance of the 'Mixture++' case is much better than any other case, even better than 'Frames++' which has similar number of images. This shows the importance of using both still images and the ample number of frames extracted from videos for improving verification performance on unconstrained faces.

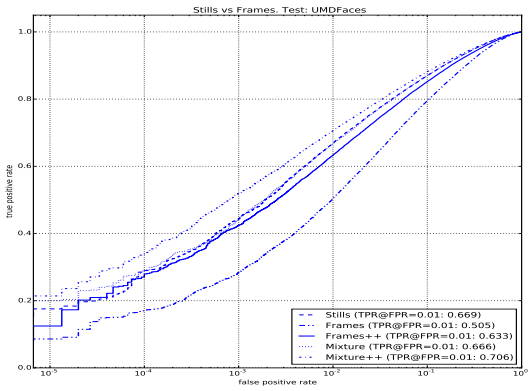


Figure 2: Verification performance of networks trained on 'Stills', 'Frames', 'Mixture', 'Frames++', 'Mixture++' and tested on UMDFaces batch-3 [2]. Note that the test set comprises of only still images. The performance of 'Stills' and 'Mixture' is very similar. However, 'Mixture++' performs best. 'Stills' performs the next best after 'Mixture++' in this case.

Also, note from figure 3 that when testing on a dataset which contains a mixture of still images and video frames [17], the performance of 'Mixture++' is the highest and 'Frames++' is the second highest. However, when testing on the UMDFaces dataset [2] which contains only images, 'Stills' performs second best after 'Mixture++' (figure 2). Similarly, when testing on the completely video-based testing set YTF [37], from figure 4, 'Mixture++' performs the best and 'Frames++' performs a bit worse than it. Also note that 'Mixture' performs better than 'Stills' and 'Frames'. Collecting millions of still images with enough variations is extremely difficult. It is much easier to collect and annotate millions of video frames. Also, using a combinations of a large number of video frames and relatively few still images gives a significant boost in performance over using only still images or video frames.

3.2. What is better: deeper or wider datasets?

For datasets with the same number of total (still) images, we call a dataset with more images per subject deeper than

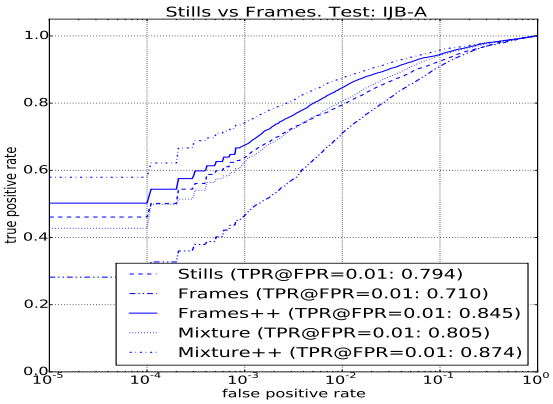


Figure 3: Verification performance of the five networks (Stills, Mixture, Frames, Mixture++, and Frames++) on IJB-A test set [17]. The IJB-A test set contains a mixture of still images and video frames. Again, the performance of 'Stills' and 'Mixed' are almost the same and 'Mixture++' is better than everything else. However, unlike figure 2, the performance of 'Frames++' is better than 'Stills'.

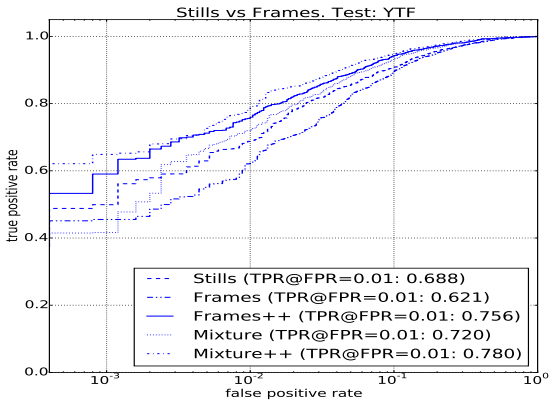


Figure 4: Verification performance of the five networks (Stills, Mixture, Frames, Mixture++, and Frames++) on YTF test set [37]. The test set contains only frames extracted from videos. Again, the performance of 'Mixture++' is better than everything else. Also, 'Mixture' performs better than 'Stills' in this case.

another dataset with fewer images per subject. We call the latter dataset wider than the prior. An example of a deep (deeper than many other still image datasets) dataset is the VGG-Face dataset [25] which has about 2.6 million images of 2,622 subjects. On the other hand CASIA-WebFace[39] can be considered a wide dataset. An extreme example of a wide dataset is the MegaFace training dataset [16, 23] which has over 670,000 subjects and only about 7 images per subject.

It is not intuitively clear whether it's better to use deeper datasets or wider for training deep networks. Given enough images, both deep and wide datasets can contain a variety of face images. Deep datasets are more varied in pose, expression, illuminations etc. On the other hand wide datasets contain large variations because of the large number of unique identities. In this section, we try to resolve the dilemma of choosing one kind of dataset over the other.

We use the UMDFaces [2], MS-Celeb-1M [7] and CASIA-WebFace [39] datasets to analyze the question. We treat batch-1 and batch-2 of UMDFaces as the training set. To explore the question of deeper vs wider datasets, we divide the training datasets into two as follows: We sort the subjects according to the number of images they have; then we start with the subject with the maximum number of images and put the subject in one set (head); we then take the subject with next highest number of images and add him/her to the head set; we continue this process till we have collected close to half the total number of images. Now we have divided each dataset into two parts. The first part (which we call 'head') contains the deeper half of the dataset. The other half is called the 'tail'. For CASIA-WebFace, the 'head' dataset contains 1,738 subjects and 247,196 images and the 'tail' set contains 8,437 subjects and 247,218 images. Similarly, the UMDFaces 'head' set has 2,142 subjects with 144,371 images and the 'tail' set has 4,092 subjects and 144,348 images.

We first train the same architecture networks on the 'head' and 'tail' sets of both CASIA-WebFace and UMDFaces. We test these networks using the protocol from section 3.1.1 on the UMDFaces batch-3 [2], IJB-A [17], and Labeled Faces in the Wild (LFW) [12] datasets. The results are shown in figures 5, 6, and 7. We note that the performance of the network trained on the 'tail' sets is better than the corresponding network trained on the 'head' sets for all three test sets. This means that, for a given number of images, it is better to have more subjects than having more images for fewer subjects.

On the other hand, if we train deeper networks, the performance of networks trained on the 'head' sets is better than the corresponding network trained on the 'tail' sets. This can be seen in figure 8 where we train ResNet-101 [9] networks on the 'head' and 'tail' sets of UMDFaces [2] and MS-Celeb-1M [7] datasets and test on the IJB-A protocol [17].

This observation is important because it can guide researchers towards better practices to follow while collecting data or selecting data for training deep networks. Data acquisition is an expensive and time consuming process and these experiments shine a light on how to obtain the maximum benefit from the investment. This is an interesting direction for future work.

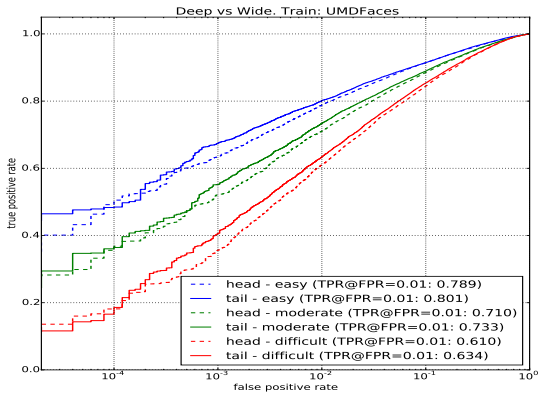


Figure 5: Training on UMDFaces [2] batch-1 and batch-2 and testing on batch-3. Solid lines represent training on the 'tail' (wide) set and dashed lines represent training on the 'head' set. We show the performance over three parts of the test dataset: easy, moderate, and hard. These parts are based on the difference in pose of the pair of images. The performance of the network trained on the 'tail' set is invariably better.

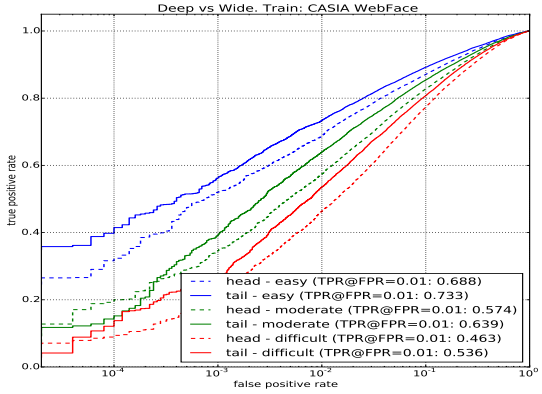
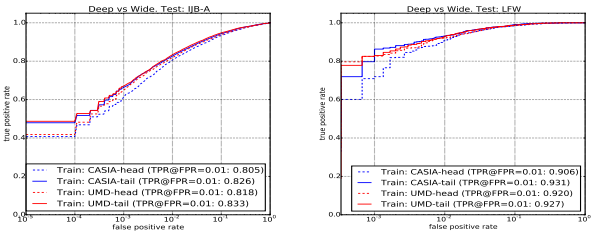


Figure 6: Verification performance of the networks trained on CASIA-WebFace [39] 'head' and 'tail' sets. We see similar trends as figure 5.

3.3. Does some amount of label noise help improve the performance of deep recognition networks?

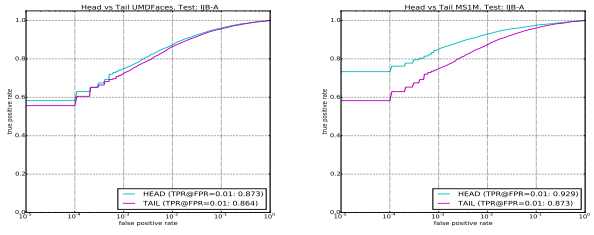
In face identification and verification research, the effect of label noise in the training set for deep networks has not been studied extensively [7, 6]. Label noise essentially means that some of the images have incorrect labels. Some [25, 7] have suggested that deep networks are robust to label noise.

We again use CASIA-WebFace [39] and UMDFaces [2] batch-1 and batch-2 for training the networks and LFW [12], IJB-A [17], UMDFaces batch-3 for evaluating the performance of these trained networks. We use the protocol explained in section 3.1.1 for evaluation.



(a) IJB-A (b) LFW

Figure 7: Performance on (a) IJB-A [17] and (b) LFW [12] of the networks trained on CASIA [39], and UMDFaces [2] 'head' and 'trail' sets. The performance of the networks trained on 'tail' are better across the range of false positive rate. (Best viewed digitally)



(a) UMDFaces (b) MS1M

Figure 8: Performance on IJB-A [17] of ResNets trained on UMDFaces [2], and MS-Celeb-1M [7] 'head' and 'trail' sets. The 'head' sets are better.

For both training datasets, we train recognition networks with $0, 2\%, 5\%$, and 10% label noise in the dataset. We would like to point out these percentages assume that the original datasets do not already contain any label noise. This assumption might not be true for many face datasets like MS-Celeb [7] and VGG-Face [25] which already contain some label noise.

Figure 9 shows the verification performance of networks trained on the CASIA-WebFace dataset for the UMDFaces test set and figure 10 shows the same for networks trained on UMDFaces dataset. There is a clear degradation in performance with increasing noise level. For both datasets, the performance of the network trained on clean data is mostly better than the performance of networks trained with even small amounts of noise. Label noise does not improve performance over clean data for face recognition. However, the difference in performance between networks trained on clean data and data with low levels of label noise is relatively low. But the percentage of noisy labels should be relatively low (less than 5%) because from figures 9 and 10, we notice that for a label noise level of 10%, the performance invariably declines.

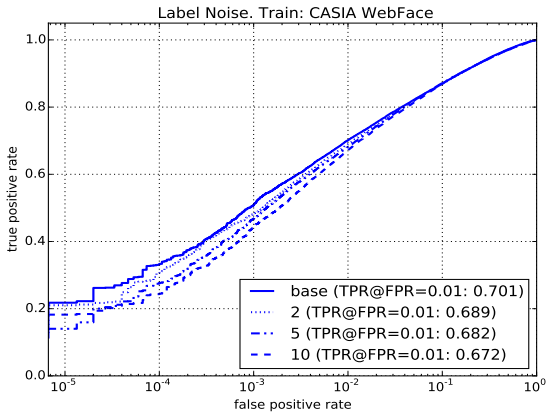


Figure 9: Performance on UMDFaces batch-3 for networks trained on CASIA WebFace [39]. Similar to figure 10 the network trained with no label noise performs best.

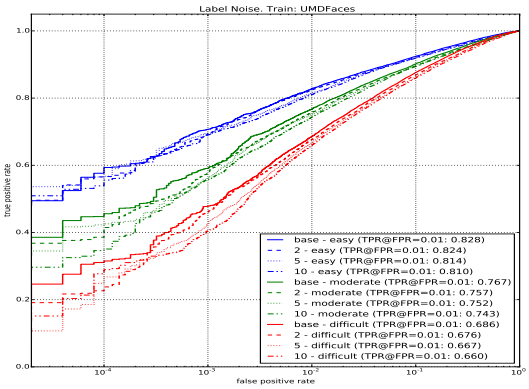
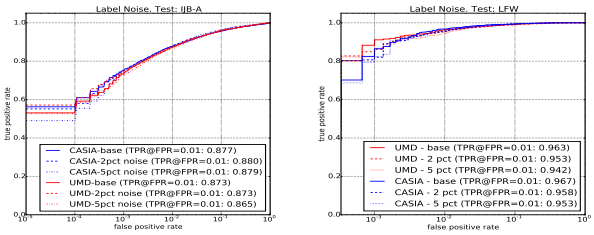


Figure 10: Verification performance on UMDFaces [2] batch-3 of deep networks trained on batch-1 and batch-2 with different noise levels. The colors represent the difficulty of test set (in terms of the difference in pose). Different line types represent different amounts of label noise added to the train set. Except for a small region in easy cases, using clean data is better than using data with label noise.

Similar trends can be seen for the LFW dataset in figure 11b. However, when testing on the IJB-A dataset [17], we notice that this observation does not hold, as shown in figure 11a. We believe that this is because the IJB-A protocol comprises of video frames which introduces another dimension of complexity for the model. Sometimes these video frames might not look like the person under consideration. We believe that such frames might be acting like a kind of label noise in the test set. That is why adding label noise to the training set might make the networks robust to such frames. However, label noise and its removal are definitely problems worthy of further research.



(a) UMDFaces (b) CASIA

Figure 11: Verification results on IJB-A [17] of networks trained on (a) UMDFaces [2], and (b) CASIA WebFace [39]. Contrary to earlier observations from figures 10 and 9, the performance on IJB-A seems to improve with adding some label noise to the train dataset. (Best viewed digitally)

3.4. Does thumbnail creation method affect performance?

Detecting [11, 27, 38, 13, 4, 36], cropping, and aligning the faces in the dataset is the first step in many face recognition pipelines. Alignment is the process of transforming a face into some canonical view. This is usually done by detecting locations of keypoints [15, 28] in the face image and then using some kind of similarity transform to transform the faces to a canonical view [25]. We refer to the images of faces obtained after cropping and/or alignment as 'thumbnails'.

We investigate whether the performance of deep recognition networks is affected by the thumbnail generation process. We compare two popular alignment techniques [28, 15] against very simple thumbnail generation techniques which only require keypoint locations and do not involve calculating any similarity transforms.

We compare three different types of thumbnails for evaluating verification performance. These are: (i) Keypoints from All-in-one CNN [28] with similarity transform alignment, (ii) DLIB keypoint detection and alignment [15], and (iii) Bounding box using keypoints from [28] without any alignment. In each case, we also study the effect of using tight thumbnails (tight crop of the face) vs loose thumbnails (including more context information). We try these methods for both training and testing and present the accuracies for the best performing cases in figures 12 and 13. We use two different datasets for training: batch-1 and batch-2 of UMDFaces [2] and CASIA-WebFace [39], and UMDFaces batch-3 for evaluating the performance of the trained networks.

All the above mentioned variations give us the following seven methods of thumbnail generation: (1) loose alignment using [28] keypoints (*aligned_uf_loose*), (2) tight alignment using [28] keypoints (*aligned_uf_tight*), (3) loose alignment using [15] keypoints (*aligned_dlib_loose*), (4)

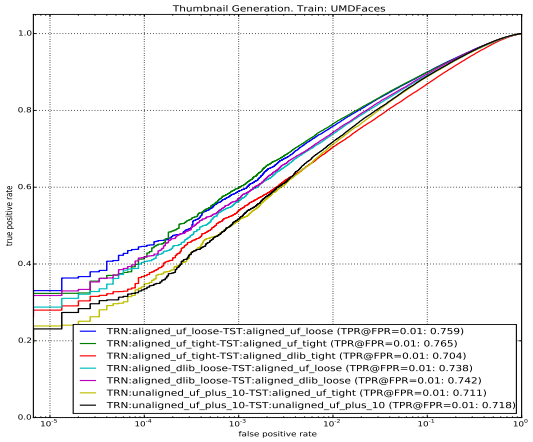


Figure 12: The performance of seven sets of train and test thumbnail generation methods. These seven were selected among all pairs of train-test pairs possible as explained in section 3.4. The training set was the UMDFaces [2] dataset in each case and the testing set was batch-3 of UMDFaces. It is clear that tightly aligning both training and testing sets using the method from [28] gives the best performance (green). (Best viewed digitally)

tight alignment using [15] keypoints (*aligned_dlib_tight*), (5) no alignment with extremely tight crops (max extent of the keypoints minus 10% of the height and width from both sides) based on keypoints obtained from [28] (*unaligned_uf_minus_10*), (6) no alignment with moderately tight crops (max extent of the keypoints) based on keypoints obtained from [28] (*unaligned_uf_tight*), and (7) no alignment but loose crops of the faces (max extent of the keypoints plus 10% of the height and width on both sides) using keypoints from [28] (*unaligned_uf_plus_10*).

We train neural networks on UMDFaces [2] and CASIA-WebFace [39] using these 7 thumbnail generation methods and test on batch-3 of UMDFaces [2] using the same 7 different thumbnail generation methods. Hence, for both training sets, we have 49 (7×7) pairs of train and test sets. For both training sets, we select the seven pairs which give the highest performance and plot them in figures 12 and 13. We note that there is a clear dependence of performance on the type of thumbnail used for training and testing. Using a good keypoint detection method and alignment procedure for both training and testing is essential for achieving good performance. Note that using a tight alignment using keypoints detected using [28] for both training and testing gives the best performance among all the cases of networks trained on UMDFaces. This pair is also a very close second among networks trained on CASIA-WebFace. As keypoint detection and alignment methods continue to improve, we expect the face verification performance to improve too.

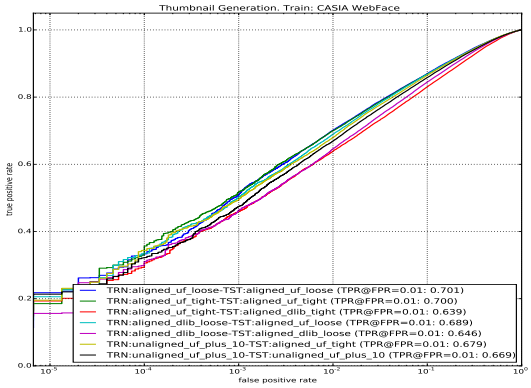


Figure 13: The performance of seven sets of train and test thumbnail generation methods with CASIA WebFace as the training set and UMDFaces batch-3 [2] as the test set. We again see that aligning both training and testing sets using [28] gives the best performance. Also, using a loose alignment gives the best performance (blue) just slightly ahead of using a tight alignment (green).

4. Conclusion

In this work we studied the effects of certain decisions about datasets and the training procedures for training deep convolutional neural networks for face verification. Carefully making these decisions is important for developing face recognition systems. This paper provides some guidelines about the decision making process. There is an abundance of video data which contain much more pose and expression variations than still images. To ensure that researchers can take advantage of this potential, we introduced a new dataset of 22,075 videos and 3,735,476 annotated frames. The importance of removing label noise from the dataset and selecting wider or deeper datasets cannot be ignored. Similarly, aligning faces using accurate keypoints during both training and testing gives a boost in performance. We hope that this work will encourage people to dig deeper into these and other decisions.

Acknowledgments

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. 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 the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] youtube-dl https://github.com/rg3/youtube-dl. online; accessed 06-march-2017. 2

[2] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv preprint arXiv:1611.01484*, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[3] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015. 3

[4] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In *European Conference on Computer Vision*, pages 122–138. Springer, 2016. 7

[5] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 1, 2

[6] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: Challenge of recognizing one million celebrities in the real world. 1, 2, 6

[7] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016. 1, 2, 5, 6

[8] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015. 1

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[10] G. Hu, X. Peng, Y. Yang, T. Hospedales, and J. Verbeek. Frankenstein: Learning deep face representations using small data. *arXiv preprint arXiv:1603.06470*, 2016. 1

[11] P. Hu and D. Ramanan. Finding tiny faces. *arXiv preprint arXiv:1612.04402*, 2016. 1, 7

[12] 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, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1, 3, 4, 5, 6

[13] V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*, 2010. 2, 7

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 3

[15] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014. 1, 7

[16] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 5

[17] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, 2015. 1, 2, 3, 4, 5, 6, 7

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4

[19] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009. 3

[20] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016. 1

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 1

[22] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 3

[23] A. Nech and I. Kemelmacher-Shlizerman. Megaface 2: 672,057 identities for face recognition. 2016. 1, 5

[24] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 343–347. IEEE, 2014. 1

[25] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015. 1, 2, 5, 6, 7

[26] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*, pages 1–8. IEEE, 2015. 1

[27] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016. 1, 7

[28] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. *arXiv preprint arXiv:1611.00851*, 2016. 1, 2, 7, 8

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1, 2

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[31] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–8. IEEE, 2016. 1, 3, 4

[32] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 1

[33] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015. 1

[34] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1489–1496, 2013. 1

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 1

[36] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 7

[37] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011. 1, 2, 4, 5

[38] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. *arXiv preprint arXiv:1511.06523*, 2015. 2, 7

[39] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 1, 2, 5, 6, 7, 8

[40] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. *arXiv preprint arXiv:1606.05413*, 2016. 1

[41] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3409–3417, 2016. 1