Analyzing and Reducing the Damage of Dataset Bias to Face Recognition with Synthetic Data

Adam Kortylewski Bernhard Egger Andreas Schneider
Thomas Gerig Andreas Morel-Forster Thomas Vetter
Department of Mathematics and Computer Science
University of Basel

Abstract

It is well known that deep learning approaches to face recognition suffer from various biases in the available training data. In this work, we demonstrate the large potential of synthetic data for analyzing and reducing the negative effects of dataset bias on deep face recognition systems. In particular we explore two complementary application areas for synthetic face images: 1) Using fully annotated synthetic face images we can study the face recognition rate as a function of interpretable parameters such as face pose. This enables us to systematically analyze the effect of different types of dataset biases on the generalization ability of neural network architectures. Our analysis reveals that deeper neural network architectures can generalize better to unseen face poses. Furthermore, our study shows that current neural network architectures cannot disentangle face pose and facial identity, which limits their generalization ability. 2) We pre-train neural networks with large-scale synthetic data that is highly variable in face pose and the number of facial identities. After a subsequent fine-tuning with real-world data, we observe that the damage of dataset bias in the real-world data is largely reduced. Furthermore, we demonstrate that the size of real-world datasets can be reduced by 75% while maintaining competitive face recognition performance. The data and software used in this work are publicly available [1].

1. Introduction

Deep face recognition systems [22, 21, 19] have achieved remarkable performances on challenging datasets, due to advances in deep learning [18] and the availability of large-scale training data [10, 13, 25]. However, training datasets for face recognition are biased regarding nuisance variables, such as the face pose or the illumination conditions, because they were mostly collected from the web. It is well known that such biases have severe negative effects on the generalization performance of machine learning systems [24, 14, 23, 17]. Therefore, the face recognition community faces two fundamental problems: 1) It is difficult to systematically analyze the effects of dataset bias on the generalization performance, since a fine-grained annotation of nuisance variables is practically unfeasible on large-scale datasets. 2) Deep face recognition systems do not generalize well across benchmarks, due to the severe sampling biases in public datasets (as illustrated in Section 4). This causes well-known problems such as a lack of diversity and fairness in face recognition [15]. It is unclear how such damages from dataset bias can be undone.

We propose to overcome both problems by leveraging synthetic face images which are generated with a parametric 3D Morphable Face Model [3, 7]. In particular, we introduce a data generator which creates synthetic face images with precise annotation of parameters that define the facial identity, such as shape and texture, but also of nuisance parameters, such as light, camera and head pose. In our experiments, we explore two application areas for synthetic images in the context of face recognition:

- **Systematic analysis of the damage from dataset bias.** We use fully annotated synthetic face images to study the face recognition rate as a function of nuisance variables such as face pose. This enables us to systematically study the effect of different types of dataset biases on the generalization ability of neural network architectures.
- **Pre-training with synthetic data.** We generate large-scale synthetic data for pre-training DCNNs and subsequently fine-tune them with real-world data. The parametric nature of the generator enables us to design the distribution of nuisances in the synthetic data such that is it highly variable in nuisance parameters that are well known to be biased in real-world datasets (such as pose and facial identity).

[1] https://github.com/unibas-gravis/parametric-face-image-generator

Based on our extensive experimental evaluation we gain several novel insights about the effects of dataset bias on the generalization ability of DCNNs at the task of face recognition: i) It is well known that DCNNs with the VGG-16 architecture can generalize better than with the AlexNet architecture at face recognition tasks. Using the presented methodology we reveal that VGG-16 outperforms AlexNet, *because* it can much better generalize to unseen face poses, although it has significantly more parameters (Section 3.2). ii) In a real world scenario, not all identities in the training data share the same distribution of face poses. We simulate this setting and observe that DCNNs cannot disentangle the facial identity from the face pose, which limits their ability to generalize from biased data (Section 3.3). iii) Using synthetic face images for pre-training, we can enhance the generalization performance of deep neural networks consistently across several benchmark datasets (Section 4.3). iv) The amount of real-world data needed to achieve competitive performance is reduced considerably (Section 4.3) after pre-training with synthetic data. Thus, offering a means to concentrate data collection efforts to less but higher quality data in terms variability.

Curiously, despite the success of 3D Morphable Face Models at facial image generation, we are not aware of any previous work that uses this effective and easily accessible approach to analyze and enhance face recognition systems.

2. Face Image Generator

We use a fully parametric generator for the synthesis of face images with detailed annotation of the most relevant nuisance transformations. Our generator is based on a 3D Morphable Model [3] of face shape, color and expression. In particular, we use the Basel Face Model 2017 (BFM-2017) [7] which is learned from 200 neutral face scans and 160 expression deformations. Natural looking, three dimensional faces with expressions can be generated by sampling from the statistical distribution of the model. In order to achieve a natural illumination in the synthetic face images, we sample the spherical harmonics illumination parameters from the Basel Illumination Prior (BIP) [5]. Using computer graphics we generate a 2D image from a 3D face, sampled from the model. We use a non-parametric background model that chooses random background textures from the data provided in the describable texture database [4]. The face image generator is built on the scalismo-faces software framework [20]. The advantage of using 3DMMs for data synthesis over related generative face models such as e.g. GANs [2, 8] is that the 3DMM provides full control over disentangled parameters that change the facial identity in the terms of shape and albedo texture as well as pose, illumination and facial expression. The proposed generator enables us to generate infinite amount of face images with detailed labeling of the most relevant sources of image variation. Example images synthesized from the generator are illustrated in Figure 1. Using the fine-grained annotation of the synthetic data enables us to systematically analyze different DCNN architectures on a common ground at the task of face recognition in the next section. Subsequently, we study how the generalization performance is affected when large-scale synthetic data is used for pre-training in Section 4.

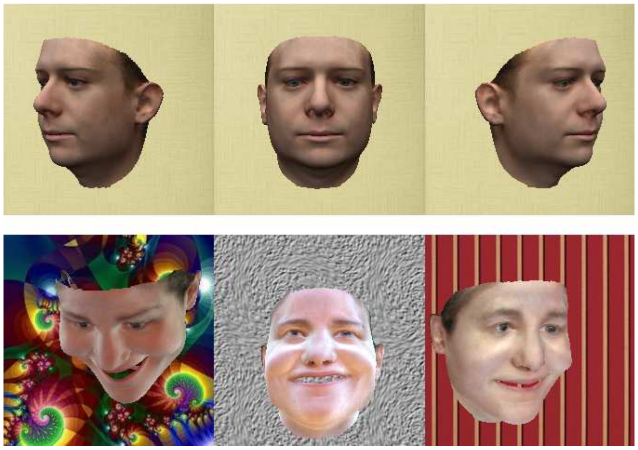


Figure 1: Synthetic face images sampled from our data generator. The facial identity in each row is the same. The top row illustrates the precise control over image parameters, where only the yaw pose is changed while all other nuisance parameters are fixed (as used in Section 3). The bottom row illustrates synthetic faces generated by randomly sampling all nuisance variables (as used in Section 4).

3. Analyzing the Damage of Dataset Bias

The fine-grained control over the image variation in the training and test data enables us to decompose the *total recognition rate (TRR)* as a function along the axis of nuisance transformations. With this tool at hand, we study how biases in the training data, in particular missing viewpoints of a face, affect the generalization of DCNNs to unseen data at test time.

3.1. Experimental Setup

Figure 2 schematically illustrates our experimental setup. We generate synthetic images of different facial identities and transform them along the axes of the nuisance transformations that we want to study (Figure 2 (I)). In this work we focus on studying the effects of biases in the face pose only. We simulate strong background variations, which are common in real world data, by sampling random textures from our empirical background model. All other nuisance parameters are fixed. We illustrate samples of the

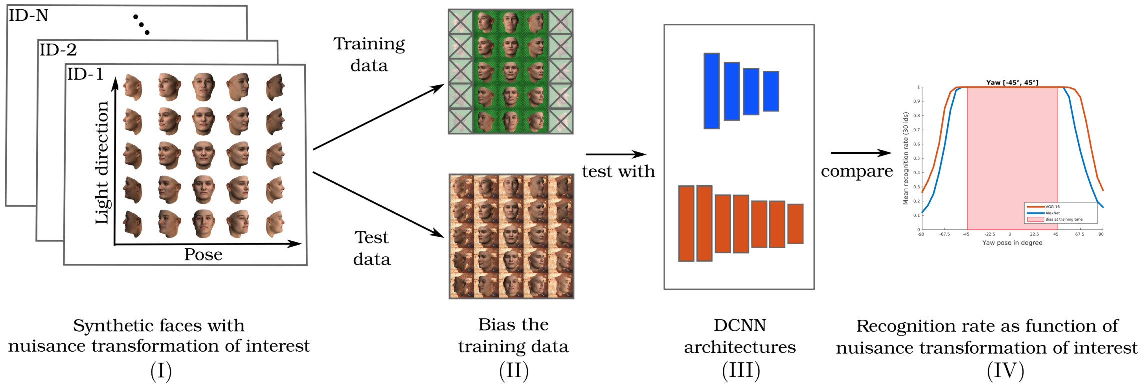


Figure 2: Experimental setup for our empirical analysis of the effect of biased training data on the generalization ability of different DCNN architectures. (I) We generate synthetic identities with a 3D Morphable Face Model and render them in different face poses. We simulate background variation by overlaying the faces on different textures. (II) We bias the training data by removing certain viewpoints from the training set. (III) We train common DCNN architectures on the biased training data. (IV) The annotation of the test data makes possible to analyze the recognition rate as a function of the face pose. It provides fine-grained information about the generalization ability of the different DCNN architectures.

face image generator with the nuisance transformations that we consider in our experiments in Figure 2. After splitting the synthetic data into a training and test set we bias the training data e.g. by removing certain face poses (Figure 2 (II)). Subsequently, we train different DCNN architectures on the biased training data (Figure 2 (III)) and evaluate how well the DCNNs generalize to the unbiased test data. The fully parametric nature of the synthetic data, allows us to evaluate the recognition rate as a function of the biased nuisance transformation (Figure 2 (IV)).

In our experiments, we focus on comparing DCNNs with a significantly diverging performance at face recognition (AlexNet and VGG-16), as our methodology makes possible to study *why* exactly one model performs better than the other. We test these networks at the task of face classification. Thus, the task is to recognize a face from an image, for which the identity is known at training time. Another common way of performing face recognition is to use the neural representation of the penultimate layer and to perform recognition via nearest neighbor in this feature space [19]. However, we focus on diagnosing the performance of DCNNs on the task that they were explicitly optimized on.

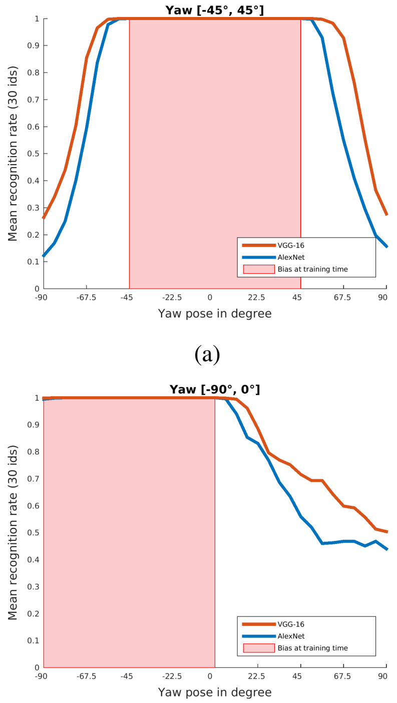
Parameter Settings. The size of the images is set to 227×227 pixels. We train the DCNNs with stochastic gradient descent (SGD) and backpropagation with the Caffe deep learning framework [12] via the Nvidia DIGITS training system. Every DCNN is trained from scratch for 30 epochs with a base learning rate of $l = 0.001$ which is multiplied every 10 epochs by $\gamma = 0.1$. We use L_2 regularization with a weight regularization parameter of $\lambda = \frac{l}{100}$. If not stated otherwise, the data is uniformly sampled across the pose and illumination axes in the specified ranges. The training data consists of 30 different identities, which we obtain by randomly sampling the shape and appearance parameter of the 3DMM. The images in the test set always reflect an unbiased sampling of the nuisance transformation that we want to study. For the yaw pose, we sample the parameter space at intervals of $\frac{\pi}{32}$ radian and for the direction of light at $\frac{\pi}{16}$ radian. Each face image is overlayed on 50 different background textures in the training as well as in the test set.

3.2. Common bias over all facial identities

In this section, we limit the range of nuisance transformations in the training data and analyze if DCNNs can generalize to the unobserved nuisance transformations. We apply the same bias to all identities in the training set (see example in Figure 5a).

EXP-1: Bias in the range of the yaw pose. In the following experiments, we limit the range of the yaw pose in the training data. The light direction is fixed to be frontal. Figure 3a illustrates the recognition performance as a function of the yaw pose, when faces in the training set are restricted to a yaw pose range of $[-45°, 45°]$. Both DCNNs achieve high recognition rates for the observed yaw poses. However, the recognition performance drops significantly when faces are outside of the observed pose range. The same generalization pattern can be observed when restricting the faces at training time to a yaw pose range of $[-90°, 0°]$ (Figure 3b). In both experiments, the VGG-16 network achieves higher overall recognition rates, *because* it generalizes better to larger unseen yaw poses.

EXP-2: Sparse sampling of the yaw pose. In Figure 4 we illustrate the effect of sampling the training data more sparsely along the axis of the yaw pose. We first bias the training set to yaw poses of $-45°$ and $45°$. VGG-16



(a)

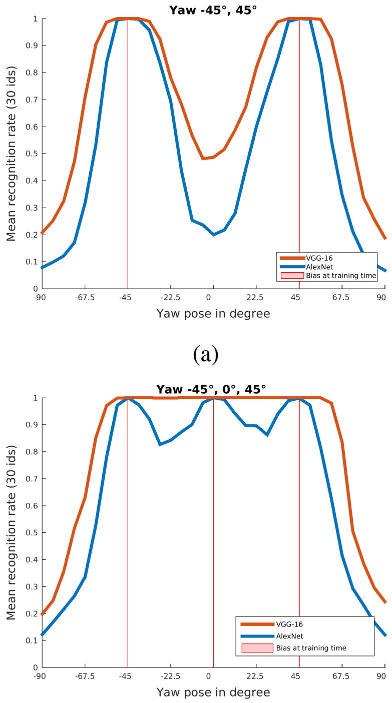
(b)

Figure 3: Effect of restricting the range of yaw poses at training time. (a) Yaw pose restricted to the range $[-45°, 45°]$. AlexNet TRR: 77.6%; VGG-16 TRR:85.9%. (b) Yaw pose restricted to the range $[-90°, 0°]$. AlexNet TRR: 81.8%; VGG-16 TRR:86.9%. In both setups the DCNNs cannot recognize faces well from previously unobserved views. VGG-16 achieves a higher TRR due to the better generalization to large unseen yaw poses.

achieves a TRR of 70.5% at test time, whereas AlexNet only achieves 51.8%. Figure 4a illustrates how these TRRs decompose as a function of the yaw pose. VGG-16 achieves constantly higher recognition rates across all poses. Most significantly, it is more than twice as good as AlexNet at recognizing frontal faces. If we add frontal faces at training time (Figure 4b) VGG-16 achieves a TRR of 81.9%, whereas AlexNet achieves 69.3%. Remarkably, VGG-16 is now able to recognize all faces correctly across the full range of $[-45°, 45°]$, whereas the recognition rates of AlexNet still drop significantly for poses in between $[-45°, 0°]$ and $[0°, 45°]$. Thus, the architecture of VGG-16 enables the DCNN to generalize well from only a few well distributed example views to other unseen views, although it has more parameters than AlexNet.

3.3. Disentanglement bias across facial identities

In the previous section, we have observed that DCNNs generalize well as soon as a nuisance transformation is sufficiently represented for *each* identity in the training. When this was not the case, the generalization performance decreased significantly. In this section, we study if DCNNs are capable of generalizing if the nuisance transformation is densely reflected in the training data across *multiple* identities. In particular, each face identity in the training is varied in a certain interval of the yaw pose. However, across all identities the full yaw pose variation is reflected. In Figure 5b we schematically illustrate how this setup compares to the one from the previous Section 3.2 (Figure 5a). We call this type of bias *disentanglement bias*, since if DCNNs are capable of disentangling the image variation induced by the yaw pose from the face identity, then they would be able to generalize well.

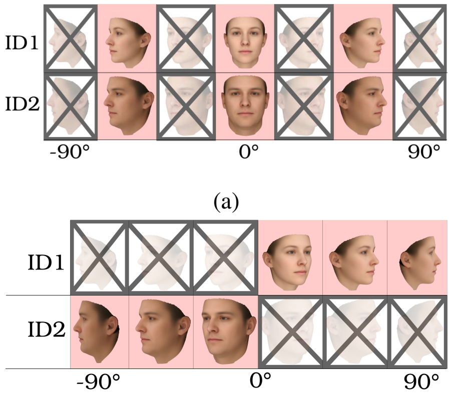


(a)

(b)

Figure 4: Effect of sparsely sampling the yaw pose of faces at training time. (a)Yaw pose sampled at $-45°$ and $45°$ (AlexNet TRR: 51.8%; VGG-16 TRR: 70.5%); VGG-16 generalizes much better to frontal poses than AlexNet. (b) Yaw pose sampled at $-45°$, $0°$ and $45°$ (AlexNet TRR: 69.3%; VGG-16 TRR: 81.9%); VGG-16 generalizes perfectly across the full range $[-45°, 45°]$, whereas AlexNet still cannot generalize in between the sampled poses.

EXP-3: Disentanglement of pose variation. In this experiment, half of the identities in the training set vary in the yaw pose range of $[-90°, 0°]$. We refer to those identities as the set Left-identities. The other half of the faces varies in the range $[0°, 90°]$ (Right-identities, Figure 5b). Figure 6



(a)

(b)

Figure 5: Different types of biases illustrated on the example of yaw pose. Faces with red background are part of the training set. (a) The same bias is applied to all the identities in the training set. Thus, the pose variation space is only partially observed. We use this setup in Section 3.2. (b) For each half of the identities an alternating half of the pose transformation is applied. Thus, the full pose transformation space is reflected in the data (Section 3.3).

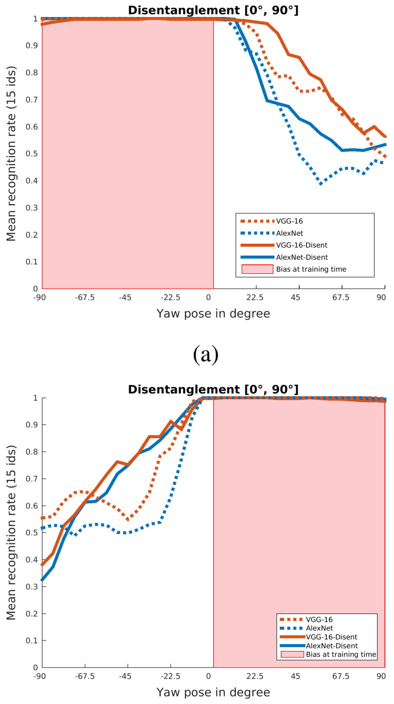
illustrates the recognition performance of DCNNs trained on the full training set. We evaluate the Left-identities and Right-identities separately (Figure 6a & Figure 6b). We observe, that the DCNNs only slightly improve compared to setup where the yaw pose range is restricted to $[-90°, 0°]$ for all identities (dotted curves). Thus, both DCNNs cannot benefit from the additional information in the training set. We conclude that this phenomenon occurs because they are not able to disentangle the image variation induced by the pose variation and the identity change.

3.4. Discussion - Analysis with Synthetic Data

Our experiments in this section demonstrate that the full control over the image variation makes possible to decompose the recognition score as a function of nuisance transformations. This enabled us to systematically analyze and compare DCNNs at the task of face recognition. In our experiements we observed the following phenomena:

Deeper networks generalize better to unseen head poses. A major reason why VGG-16 outperforms AlexNet at face recognition is that it can generalize better to faces in previously unseen face poses (Section 3.2).

Deep networks cannot disentangle face pose from facial identity. A major limitation of the analyzed DCNN architectures is that they have severe difficulties to generalize when facial identities do not share the same pose variation (Section 3.3). Thus, deep networks cannot disentangle well the image variation caused by changes in the face pose from the one induced by changes in the facial identity.



(a)

(b)

Figure 6: Testing disentanglement ability of DCNNs. Dotted lines: DCNNs trained on a biased yaw pose (illustrated in Figure 5a). Solid lines: Disentanglement setup (illustrated in Figure 5b). (a) Left-Identities with biased yaw pose of $[-90°, 0°]$. (b) Right-Identities with biased yaw pose of $[0°, 90°]$. DCNNs cannot make use of the additional information about the pose transformation which is present in the data in the disentanglement setup.

4. Reducing the Damage of Dataset Bias

In this section, we study the impact on the generalization performance when using large-scale synthetic data for pretraining of deep face recognition systems.

4.1. Experimental Setup

Our face recognition experiments are based on the publicly available OpenFace framework [1]. For face detection and alignment we use a publicly available multi-task CNN [2] [26]. In case the face detection fails, we use the face boxes as defined in the individual datasets [3]. We train the FaceNet-NN4 architecture that was originally proposed by Schroff et al. [21] with the vanilla setting, as provided in the OpenFace framework. The aligned images are scaled to 96×96 pixels.

[2] https://github.com/kpzhang93/mtcnn_face_detection_alignment

[3] For LFW and IJB-A these face boxes are provided in the dataset, for Multi-PIE we use the annotations provided in [6].

Face Recognition			
Datasets	**Multi-PIE**	**LFW**	**IJB-A**
Metric	Accuracy	Accuracy	TAR
SYN-only	88.9	80.1	62.5
Real-100%	91.2	94.1	86.8
+ Primed	**93.3**	**95.8**	**90.6**
Real-25%	83.6	89.1	71.3
+ Primed	**91.3**	**93.6**	**85.0**
Real-10%	81.7	85.1	66.2
+ Primed	**91.3**	**91.8**	**83.4**

Table 1: Face recognition performance on the CMU-Multi-PIE, LFW and IJB-A benchmarks. We compare models trained on synthetic face images (SYN-only) to models trained on different sized subsets of the Casia dataset (Real-$\{10\%, 25\%, 100\%\}$). We denote primed models that were fine-tuned on real-world data by "+ Primed" below the corresponding real-world data only result. We measure performance in terms of recognition accuracy and the true acceptance rate (TAR) at false acceptance rate $FAR = 0.1$.

The triplet loss is trained with batches of 20 identities and 15 sample images per identity for 200 epochs.

The real-world training data for face recognition is sampled from the cleaned Casia WebFace dataset [25], which comprises 455,594 images of 10,575 different identities. From this dataset, we remove the 27 identities which overlap with the IJB-A dataset. For testing the generalization performance we use: CMU-Multi-PIE [9], LFW [11] and IJB-A [16]. We measure the distance between two face images as the cosine distance between their 128-dimensional feature embeddings from the last layer of the FaceNet model. We do not perform any dataset adaptation, thus we test the most challenging face recognition protocol with only *unrestricted, labeled outside data*.

Synthetic face image generation. The synthetic face images used for training are generated as described in Section 2. The head pose is sampled according to a uniform pose distribution on the yaw, pitch and roll angles in the respective ranges $r_{yaw} = [-90°, 90°]$, $r_{pitch} = [-30°, 30°]$ and $r_{roll} = [-15°, 15°]$. For face recognition, we generate one million face images with $20K$ different identities and 100 example images per identity.

4.2. Dataset Bias in Real and Synthetic Data

When training with the full set of real-world data (Real-100%) we observe that the distribution of the training data is similar to some benchmark datasets (e.g. LFW), while it is a lot less so for others, such as IJB-A (Table 1). When training with synthetic data only (SYN-only), we observe that for the CMU-Multi-PIE benchmark the performance is similar to that of a deep network trained with real-world data . This suggests that our synthetic face images can well represent the facial appearance in constrained visual environments. However, on the benchmarks of LFW and IJB-A the SYN-only model performs worse when compared to a network trained with the full real-world dataset.

Note that datasets with synthetic and real face images have different kinds of biases. In real data some facial properties such as head pose, illumination or facial expression are difficult to annotate and therefore cannot be taken into account when collecting data. In synthetic data, these properties can be modeled very well and thus can be sampled extensively, however, other characteristics of faces are currently not modeled with parametric face models. These include e.g. facial hair, partial occlusion, the mouth interior or detailed skin textures. In the following section, we explore the potential of combining both types of data and their complimentary biases.

4.3. Priming with Synthetic Data Enhances Performance

We fine-tune the SYN-only model with different subsets (10%, 25%, 100%) of the real-world training data. In this way, the synthetic data will prime the model towards the target facial image analysis task, enabling the model to leverage the information in the real-world data more efficiently during the fine-tuning process. The performance of the primed models is denoted as "+ Primed" in Table 1. The primed models considerably outperform the unprimed models at face recognition. Interestingly, even when fine-tuning with the full real-world datasets the models still have an enhanced performance compared to the unprimed models.

The real-to-virtual gap can almost be closed with 25% of the real data. Remarkably, priming with synthetic data leads to a performance increase across *all* benchmarks, even though the individual datasets have very different imaging characteristics. Note that we do not perform any dataset adaptation in our experiments.

4.4. Discussion - Priming with Synthetic Data

In this section, we observed the following phenomena when using synthetic face images for priming deep face recognition systems:

Enhanced generalization performance. Priming with synthetic data followed by fine-tuning with real-world data enhances the generalization performance consistently across all benchmark datasets compared to training with real-world data only.

Enhanced data efficiency. Using our priming approach the number of real-world data needed to achieve competitive performance at face recognition was reduced by 75% (Section 4.3).

5. Conclusion

In this work, we have demonstrated the large potential of using synthetic face images to systematically analyze and reduce the damage of dataset bias to face recognition.

In particular, we studied the effect of dataset bias on the generalization performance of different and DCNN architectures and observed that deeper neural network architectures generalize better to unseen poses. Furthermore, a major limitation of current neural network architectures is that they cannot disentangle facial identity from face pose, which severely limits their ability do generalize from biased training data.

When using synthetic data for pre-training deep face recognition systems, we observe that the damage from dataset bias in real-world data can be largely reduced. In addition, we showed that the number of real-world face images needed to achieve a competitive face recognition performance can be reduced by 75% using synthetic data.

In summary, our experimental results suggest that novel insights about the damage of dataset bias to face recognition can be gained from systematically analyzing deep face recognition systems with synthetic data. Furthermore, deep face recognition systems should be pre-trained with synthetic data to enhance their generalization performance, in particular when the facial properties in the training and test data are expected to be distributed differently.

Acknowledgements: A.K. is supported by the SNSF grant P2BSP2_181713. We gratefully acknowledge the support of NVIDIA with the donation of a Titan Xp.

References

[1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] D. Berthelot, T. Schumm, and L. Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH'99 Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press, 1999.

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] B. Egger, S. Schoenborn, A. Schneider, A. Kortylewski, A. Morel-Forster, C. Blumer, and T. Vetter. Occlusion-aware 3d morphable models and an illumination prior for face image analysis. *International Journal of Computer Vision*, 2018.

[6] L. El Shafey, C. McCool, R. Wallace, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[7] T. Gerig, A. Forster, C. Blumer, B. Egger, M. Luethi, S. Schoenborn, and T. Vetter. Morphable face models - an open framework. *CoRR*, abs/1709.08398, 2017.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 2010.

[10] 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.

[11] 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.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] 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*, pages 4873–4882, 2016.

[14] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, 2012.

[15] B. F. Klare, M. J. Burge, J. C. Klontz, R. W. V. Bruegge, and A. K. Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

[16] 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.

[17] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster, and T. Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2093–2102, 2018.

[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.

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

[20] S. Schoenborn, A. Schneider, A. Forster, and B. Egger. Scalismo Faces. https://github.com/unibas-gravis/scalismo-faces/, 2016. [Online; accessed 01-November-2017].

[21] 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.

[22] 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.

[23] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer, 2017.

[24] A. Torralba and A. A. Efros. Unbiased look at dataset bias. 2011.

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

[26] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2017.