Residual vs. Inception vs. Classical Networks for Low-Resolution Face Recognition

Christian Herrmann[1,2](✉), Dieter Willersinn[2], and Jürgen Beyerer[1,2]

[1] Vision and Fusion Lab, Karlsruhe Institute of Technology KIT,
Karlsruhe, Germany
[2] Fraunhofer IOSB, Karlsruhe, Germany
{christian.herrmann,dieter.willersinn,
juergen.beyerer}@iosb.fraunhofer.de

Abstract. When analyzing surveillance footage, low-resolution face recognition is still a challenging task. While high-resolution face recognition experienced impressive improvements by Convolutional Neural Network (CNN) approaches, the benefit to low-resolution face recognition remains unclear as only few work has been done in this area. This paper adapts three popular high-resolution CNN designs to the low-resolution (LR) domain to find the most suitable architecture. Namely, the classical AlexNet/VGG architecture, Google's inception architecture and Microsoft's residual architecture are considered. While the inception and residual concept have been proven to be useful for very deep networks, it is shown in our case that shallower networks than for high-resolution recognition are sufficient. This leads to an advantage of the classical network architecture. Final evaluation on a downscaled version of the public YouTube Faces Database indicates a comparable performance to the high-resolution domain. Results with faces extracted from the SoBiS surveillance dataset indicate a superior performance of the trained networks in the LR domain.

1 Introduction

Forensic analysis of video data can help to solve crimes and identify the offenders. The struggle to evaluate increasing amounts of video footage can be addressed by automated solutions. A key component for automated video analysis is a robust face recognition system for person identification. Recent Convolutional Neural Network (CNN) approaches [19,21,28] which lead to significant performance improvements on this task, mainly came from internet companies such as Google or Facebook. Consequently, the addressed target domain are high quality face shots such as celebrity photos, selfies or personal profile photos. To address the low-resolution (LR) surveillance domain, at least an adaptation of such networks is required.

Several different types of network architectures are successfully applied for CNNs. Here, we select three different state-of-the-art architecture types (inception [21], residual [5] and classical [19]) and evaluate their suitability for low-resolution (LR) face recognition. An analysis of the respective architectures is performed to identify the necessary adjustments and potential bottlenecks of

P. Sharma and F.M. Bianchi (Eds.): SCIA 2017, Part II, LNCS 10270, pp. 377–388, 2017.
DOI: 10.1007/978-3-319-59129-2_32

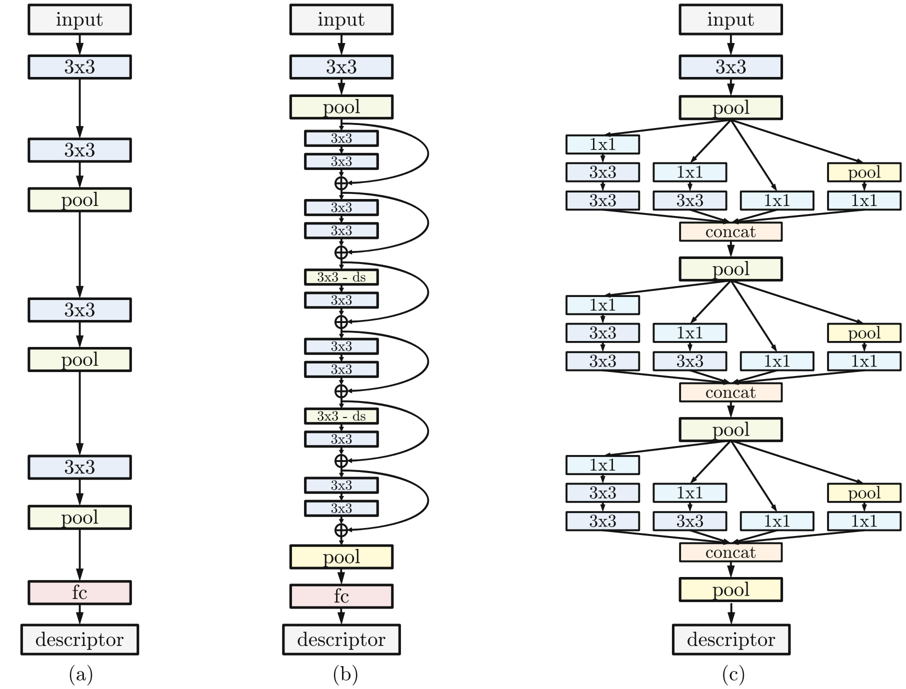

Fig. 1. Adapted LR networks for different architecture types: classical (a), residual (b) and inception (c). Green background denotes downsampling layers. (Color figure online)

the networks for the LR task. According to the analysis, each architecture is implemented with regard to the LR domain. After a systematical optimization of the meta-parameters, the final evaluation is performed on a LR version of the YouTube Faces Database (YTF) [30] as well as surveillance domain faces extracted from the SoBiS person identification database [22].

2 Related Work

Face recognition is a common topic in computer vision and recent state-of-the-art solutions for high-resolution (HR) faces are usually all based on CNNs [3,14,15,19,21,28,29]. While groups from companies usually train on their own very large scale datasets containing up to several hundred million face images [21,28], other groups are limited to smaller public datasets [4,13,17–19,31,32] usually containing less or slightly above one million face images. Data augmentation strategies [15] can compensate for this but might also introduce unwanted effects.

In comparison to HR approaches, recent LR face recognition strategies are usually still based on conventional strategies consisting of a combination of local features and learned representations such as metric learning [16], dictionary learning [23] or manifold learning [11]. Currently, only few attempts to apply CNN approaches exist in this domain [7,8]. In summary, this gives the impression that LR face recognition is still moving towards the deep learning age.

Especially, unlike the HR face recognition task where it is possible to fine tune pre-trained networks [19], a LR face recognition network has to be trained from scratch due to the lack of general LR networks. One key choice which has to be made is the architecture type of the network. The most recent CNN architectures are usually first developed and proposed for the ImageNet challenge [2]. The three currently most distinctive and widespread architecture types are the inception architecure [26,27], the residual architecture [5,6] and the rather classically designed VGG architecture [25], which is quite similar to the well known AlexNet [12]. In the following sections, we will analyze and adjust these three architecture types to the LR scenario and finally compare their performance.

3 Network Architecture

When designing a CNN for LR face recognition, several issues have to be considered. For the ease of presentation, we focus on a face resolution of 32×32 pixels.

The main issue when adapting the different architectures to LR scenarios are the downsampling layers usually implemented as pooling layers. They group spatial information by propagating the average or maximum of a $k \times k$ region. For the common choice of $k = 2$ this involves a downsampling by a factor of two. If some spatial information should be kept in the feature maps, a maximum of 4 pooling layers is acceptable, for 32×32 pixel input size. The 5th pooling layer would condense the remaining 2×2 feature map into a 1×1 feature map and destroy the respective spatial information.

This conflicts one general rule of thumb for creating CNNs stating that deeper networks are preferred over wider networks. When depth is limited by the maximum number of possible downsampling steps, it will be necessary to compromise by using wider networks than for HR applications.

3.1 Classical Architecture

This traditional philosophy became popular with the famous AlexNet [12] and was extended with minor adaptions in the VGG Face network [19]. Some guidelines how to transfer this architecture to the LR task are given in [8].

Conceptually, the beginning of the network consists of alternating convolutional and pooling layers as shown by Fig. 2a. After these layers, a set of fully connected layers is appended to classify the resulting feature maps of the first part.

As motivated by the authors of the VGG network [25], in this architecture, consecutive convolutional layers can be understood as a replacement of a single convolutional layer with a larger filter size. This means for example that two 3×3 convolutional layers are a replacement for one 5×5 convolutional layer.

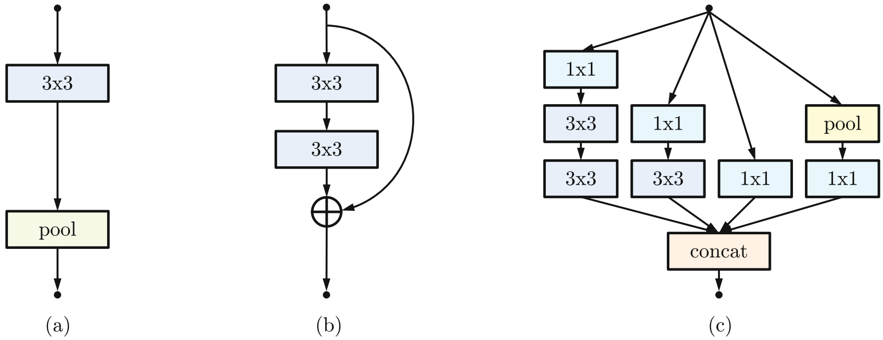

Fig. 2. Key components for each architecture type. Alternating convolutional and pooling layers for classical type (a), residual block (b) and inception module (c).

Following this motivation, the number of layers in such a network has to remain limited for LR applications because filter sizes have to represent the small content size in the LR image. Large filter sizes in the shape of many consecutive convolutional layers are unnecessary or even counterproductive.

Design choices for this architecture type include the number of convolutional, pooling and fully connected layers as well as the layer meta parameters itself. These include the number of filters per convolutional layer C, kernel size $P \times P$ of pooling layers or the number of neurons N in fully connected layers. To ease optimization, the basic structure of the network starts with a 3×3 convolutional layer and continues with a varying number of groups each consisting of one 3×3 convolutional layer and one pooling layer.

3.2 Residual Architecture

The main difference to the classical architecture is a kind of bypass of two convolutional layers (Fig. 2b). The benefit according to He et al. [5] is the better trainability for very deep networks. Instead of learning a function mapping of the input x to the output $F(x)$ where F represents the learned function, this strategy has to learn only a small offset or residuum $F(x)$ of the identity, leading to the output $F(x) + x$. Using this trick, this architecture type currently allows the deepest networks with about 1000 layers in certain applications [6]. Because the potentially lower number of layers in the application of this paper limits training time, the full residual blocks instead of the reduced bottleneck ones are employed. A specialty of the residual architecture are the missing pooling layers for downsampling. This is instead performed by specific convolutional layers which have a stride of 2 resulting also in a downsampling by factor 2. Once again, the amount of these special layers limits network depth for LR scenarios. Calling a fixed set of consecutive residual blocks a group, one of these downsampling layers is inserted at the beginning of each group.

Key design choices for this architecture include the number of residual groups, their sizes, as well as the number of trailing fully connected layers, besides the layer meta parameters itself.

3.3 Inception Architecture

The key concept of the inception architecture is a kind of meta layer called inception module [26,27] shown by Fig. 2c. It includes several parallel data processing paths which are motivated by multi-scale processing. Efficient usage of computing resources is implemented by 1×1 convolutional layers which reduce data dimension.

Besides the layer meta parameters, the key design choice is the number of inception modules and trailing fully connected layers. Filter numbers within an inception module are fixed to a ratio of 1:4:2:1 for the double 3×3, single 3×3, 1×1 and pooling path respectively. The filter number ratio between 1×1 and consecutive 3×3 layers is 1:2. Following the same argumentation about filter sizes as for the classical architecture, we add a pooling layer after each inception module to avoid overly large filter sizes and call the combination of inception module and pooling layer a group.

3.4 General Configuration and Training Setup

As shown by the examples in Fig. 1, the basic structure of each network begins with a 3×3 convolutional layer followed by a pooling layer in case of residual and inception architectures. Afterwards, a modifiable number of layer groups of the respective architecture kind is added. The number of convolutional filters is doubled after each group in the network. In all cases, downsampling is performed between layer groups. The networks end with a varying number of fully connected layers in front of the output layer which includes the descriptor.

All networks are trained using a Siamese setup which has two key advantages compared with a conventional softmax classification strategy. First, the descriptor dimension, which equals the number of neurons in the output layer, is independent of the number of person identities in the dataset and can be chosen arbitrarily. In practice, this allows smaller descriptors. Second, it allows to combine several datasets without any effort because no consistent identity labels between datasets are required. As loss function, the max-margin hinge loss proposed in [8] is employed.

With this setup, the network can be understood as a projection from the image space into a discriminative feature space with respect to face recognition. The dimension of this feature space, which equals the target descriptor size, is set to 128 motivated by a number of further approaches [8,21,24]. Batch normalization [9] is incorporated in all cases.

4 Experiments

Exploiting the advantage of the Siamese setup, the networks are trained on a combination of several large-scale face datasets including Celebrity-1000 [13], FaceScrub [18], MegaFace [17], MSRA [32], TV Collection (TVC) [8], and VGG Face Dataset [19]. This results in about 9M training images. The recent MS-Celeb-1M [4] as well as the CASIA WebFace [31] databases are omitted, because we found that they do not improve the results. All networks are trained from scratch using the Caffe framework [10] and Nvidia Titan X GPUs. Evaluation is performed on a downscaled version of the YouTube Faces Database (YTF) [30] (1,595 persons, 3,425 sequences) as well as faces collected from the SoBiS surveillance dataset [22]. Faces from the SoBiS dataset are extracted by a Viola-Jones based face tracker and registered by eye locations [20]. This results in 1,559 face sequences from 45 persons.

4.1 Architecture Optimization

Some of the training datasets include identities also present in the YTF database used for the final evaluation. These are removed from the training set and serve as validation set in this section. Table 1 lists all datasets including the relevant split sizes. Due to memory limitations, architecture optimization in this section is performed without the MegaFace and VGG Face dataset. After determination of a potential parameter optimum for each architecture type, a systematic optimization is performed in each case. The batch size is set to 100 for the majority of the networks and reduced to 20 for the largest networks. Each batch contains samples from all datasets. Table 2 shows selected results for some key parameters. The filter number C is given for the first 3×3 convolutional layer.

Having a detailed look at the impact of the different meta parameters, one can note a few things in Table 2. Regarding the number of layer groups, either 3 or 4 perform best (case 1–3). Larger differences were observed for the number of fully connected layers at the end of the network (case 11–13). At least one is required in the classical, and exactly one in the residual architecture. The results for the inception architecture are inconclusive with no or two fully connected layers performing comparably and, in the sense of reducing parameters,

Table 1. Training and validation datasets. Image datasets have the same number of images and sequences (each image can be understood as single sequence), video datasets have multiple images per sequence. Differences to official dataset sizes might occur in case of images being no longer downloadable from internet links.

Dataset name	Train			Validation		
	#images	#sequences	#persons	#images	#sequences	#persons
Celebrities-1000 [13]	2, 117, 837	145, 751	930	210, 154	13, 981	70
FaceScrub [18]	51, 162	51, 162	451	10, 299	10, 299	79
MegaFace [17]	4, 741, 425	4, 741, 425	672, 957	–	–	–
MSRA [32]	163, 018	163, 018	1, 372	39, 704	39, 704	208
TV collection [8]	1, 151, 545	15, 427	604	–	–	–
VGG face [19]	834, 375	834, 375	2, 558	–	–	–
Combination	9, 059, 362	5, 951, 158	678, 872	260, 157	63, 984	357

Table 2. Optimization of the three architectures on the validation set at 32×32 pixels face size. Baseline values are indicated. Results are mean accuracy and standard deviation for a 10-fold cross-validation. Blanked settings were omitted due to limited GPU memory.

Parameter	Case	Value	Classical		Residual		Inception	
			Accuracy	Std	Accuracy	Std	Accuracy	Std
#groups	1	2	0.771	0.013	0.664	0.006	0.754	0.011
	2	3	*0.776*	0.010	*0.758*	0.009	*0.772*	0.013
	3	4	0.789	0.010	0.770	0.009	0.657	0.008
#filters C	4	64	0.687	0.014	*0.758*	0.009	0.728	0.008
	5	128	0.771	0.013	0.762	0.007	0.755	0.015
	6	192	0.784	0.018	0.760	0.012	*0.772*	0.013
	7	256	*0.776*	0.010	0.762	0.009	0.778	0.011
	8	384	0.784	0.009	-	-	0.737	0.112
	9	512	0.786	0.013	-	-	-	-
	10	768	0.795	0.012	-	-	-	-
#fully connected layers	11	0	0.767	0.010	0.719	0.004	*0.772*	0.013
	12	1	*0.776*	0.010	*0.758*	0.009	0.732	0.013
	13	2	0.779	0.012	0.641	0.009	0.777	0.008
#neurons per fully connected layer N	14	1024	0.782	0.012	0.761	0.010	0.732	0.013
	15	2048	*0.776*	0.010	*0.758*	0.009	0.693	0.015
	16	4096	0.781	0.011	0.765	0.006	0.636	0.022
	17	8192	0.759	0.014	0.700	0.010	0.581	0.018

it is preferred to choose none. The different impact of the fully connected layers can be explained by the varying capabilities of the respective architecture blocks in front. The results for the number of filters in the convolutional layers behave as expected (case 4–10). In each case, increasing the number of filters improves the results up to a range where saturation is observed. The upper limit is caused by GPU memory in all cases. The number of neurons in any fully connected layer is rather irrelevant in the tested range, except for inception where performance decreases with increasing number of neurons (case 14–17). All in all, the classical architecture performs best with the inception architecture coming second. This indicates that architectures allowing deeper networks are inferior for LR applications compared to the classical concept. It can be suspected that the LR images contain too few information to feed a very deep network. Note that the differences in validation accuracy near the optimum are mostly below the measured standard deviation. So from a statistical viewpoint, the measurement noise on the accuracy justifies the choice of a parameter setting which seems

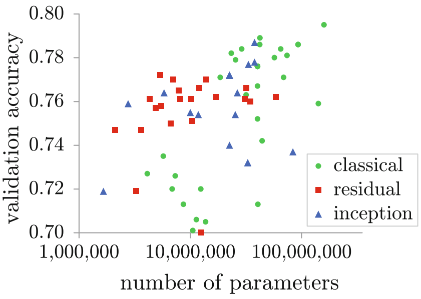

Fig. 3. Validation accuracy vs. number of network parameters.

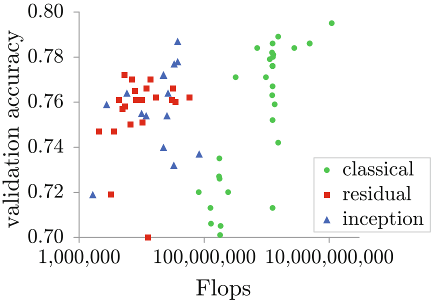

Fig. 4. Validation accuracy vs. network Flops

Table 3. Final network configuration and properties for each architecture type.

	Classical	Residual	Inception
Configuration (cases)	2,9,12,16	2,5,12,16	2,7,11
Validation accuracy	0.786	0.766	0.787
Parameters	74.1 M	32.0 M	29.2 M
Flops	1,279 G	512 G	986 G
Prediction time (ms)	2.1	4.2	6.4
Memory footprint (MB)	338	230	239

to be slightly off the maximum. This becomes relevant if runtime and memory consumption has to be considered along with the performance.

One of the main differences between the classical architecture and the modern residual and inception one is the number of parameters necessary to achieve a certain performance. The modern architectures require less parameters to achieve a comparable performance as shown by Fig. 3 which includes all trained networks. Similarly, Fig. 4 shows a comparison between performance and necessary computational power for all the trained networks. But the stated number of Flops is only a theoretical value. In practice, execution times increase significantly with network depth due to relatively higher memory usage by the large amount of intermediate layers.

Table 3 shows the properties for the best network of each architecture including measured execution times on a Titan X. Note that despite having the most Flops, the classical network's forward pass is the fastest due to less required memory operations. These three networks will serve for final training and evaluation.

4.2 In-the-Wild Results

Final training is performed using the best setup for each architecture type as denoted by Table 3 and visualized by Fig. 1 with all training data listed in Table 1

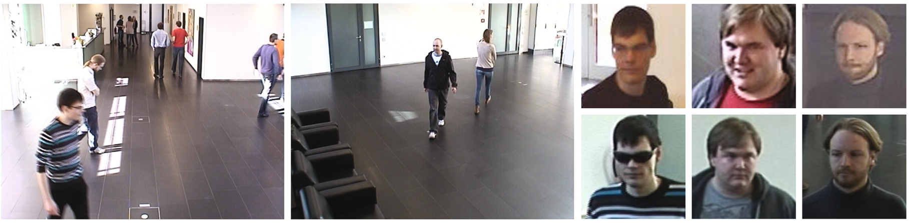

Fig. 5. Example scenes and faces from the SoBiS Face dataset.

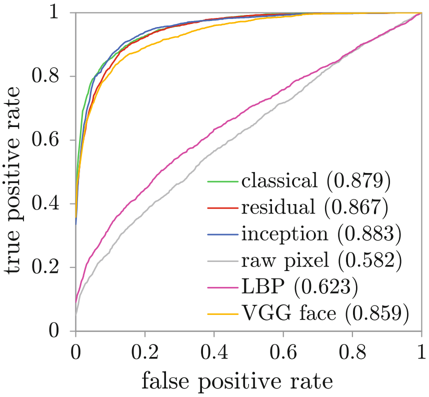

Fig. 6. ROC curve and accuracy (in brackets) comparison for low-resolution YTF dataset.

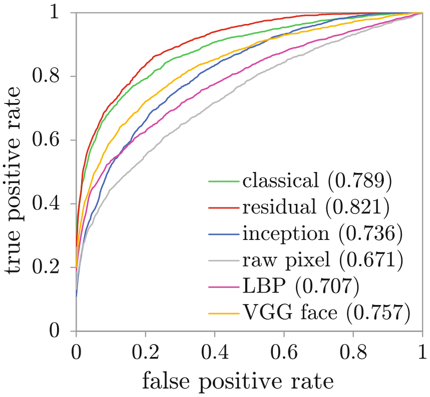

Fig. 7. ROC curve and accuracy (in brackets) comparison for SoBiS Face dataset.

until convergence. For comparison, raw pixel, local binary patterns (LBP) [1], and the HR VGG Face descriptors [19] are employed. The appropriate vector distance is combined with each descriptor: Hellinger distance for LBP because of its histogram character, cosine distance for raw pixel and VGG Face because of the softmax training setup and Euclidean distance for LrfNet and the trained descriptors because the loss function was chosen to minimize this distance. Aggregation of multiple frames in a consecutive face sequence is performed by averaging the respective face descriptors.

A 10-fold cross-validation is performed on both the YTF and SoBiS Face dataset (Fig. 5). The results are reported for all face descriptors as ROC curves and the respective classification accuracy in Figs. 6 and 7. While all CNN-based descriptors significantly outperform the LBP descriptor in the case of higher quality data represented by the YTF dataset, the high-quality trained VGG Face descriptor performs nearly as bad as LBP in the case of the SoBiS surveillance data. The proposed face descriptors with exception of the inception one outperform the VGG Face descriptor by a significant margin in this case. We are unsure about the reason for the significant performance drop of the inception architecture compared to the high-quality YTF data. Potential explanations include a too strong focus on fine grained details by the multi-scale inception blocks or domain overfitting by this powerful architecture on the training data which is more similar to the YTF than the SoBis data. Comparing the results on both datasets, it can be observed that the order of the architectures is inverse in the YTF and SoBiS experiments. This supports the overfitting hypothesis where the residual architecture generalizes best, classical coming second and the inception being the worst in this respect. All in all, the achieved accuracy of 0.883 on YTF is near the HR accuracy 0.916 of VGG Face on the same dataset.

5 Discussion and Conclusion

All in all, the analysis of the three CNN architecture concepts classical, residual and inception allowed their adaptation to the LR domain. The optimization and evaluation for LR face recognition revealed that in this case the classical architecture performs best. While this result is inverse to the architecture capabilities shown in literature on image categorization and HR face recognition, it can be explained by the low image resolution. The spatial resolution limits the amount of downsampling layers in the network which makes it unreasonable to create very deep networks where the capabilities of the residual and inception architecture unfold. Training dedicated LR face descriptors was proven to yield better results than the application of a HR face descriptor such as VGG Face. Especially when surveillance data has to be analyzed, the proposed LR face descriptors outperform further descriptors significantly.

References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
3. Ding, C., Tao, D.: Trunk-Branch Ensemble Convolutional Neural Networks for Video-based Face Recognition. arXiv preprint arXiv:1607.05427 (2016)
4. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: challenge of recognizing one million celebrities in the real world. In: Imaging and Multimedia Analytics in a Web and Mobile World (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. arXiv preprint arXiv:1603.05027 (2016)
7. Herrmann, C., Willersinn, D., Beyerer, J.: Low-quality video face recognition with deep networks and polygonal chain distance. In: Digital Image Computing: Techniques and Applications, pp. 1–7. IEEE (2016)
8. Herrmann, C., Willersinn, D., Beyerer, J.: Low-resolution convolutional neural networks for video face recognition. In: Advanced Video and Signal Based Surveillance. IEEE (2016)

9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (2015)
10. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093 (2014)
11. Jiang, J., Hu, R., Wang, Z., Cai, Z.: CDMMA: coupled discriminant multi-manifold analysis for matching low-resolution face images. Sig. Process. **124**, 162–172 (2016)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Liu, L., Zhang, L., Liu, H., Yan, S.: Toward large-population face identification in unconstrained videos. IEEE Trans. Circuits Syst. Video Technol. **24**(11), 1874–1884 (2014)
14. Liu, X., Kan, M., Wu, W., Shan, S., Chen, X.: VIPLFaceNet: An Open Source Deep Face Recognition SDK. arXiv preprint arXiv:1609.03892 (2016)
15. Masi, I., Tran, A.T., Leksut, J.T., Hassner, T., Medioni, G.: Do we really need to collect millions of faces for effective face recognition? arXiv preprint arXiv:1603.07057 (2016)
16. Mudunuri, S.P., Biswas, S.: Low resolution face recognition across variations in pose and illumination. IEEE Trans. Pattern Anal. Mach. Intell. **38**(5), 1034–1040 (2016)
17. Nech, A., Kemelmacher-Shlizerman, I.: Megaface 2: 672,057 Identities for Face Recognition (2016)
18. Ng, H.W., Winkler, S.: A data-driven approach to cleaning large face datasets. In: International Conference on Image Processing, pp. 343–347. IEEE (2014)
19. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference, vol. 1(3), p. 6 (2015)
20. Qu, C., Gao, H., Monari, E., Beyerer, J., Thiran, J.P.: Towards robust cascaded regression for face alignment in the wild. In: Computer Vision and Pattern Recognition Workshops (2015)
21. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Computer Vision and Pattern Recognition, pp. 815–823 (2015)
22. Schumann, A., Monari, E.: A soft-biometrics dataset for person tracking and re-identification. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), August 2014
23. Shekhar, S., Patel, V.M., Chellappa, R.: Synthesis-based robust low resolution face recognition. IEEE Trans. Image Process. (under review) (2014)
24. Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher vector faces in the wild. In: British Machine Vision Conference, vol. 1, p. 7 (2013)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition, pp. 1–9 (2015)
27. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567 (2015)

28. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
29. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). doi:10.1007/978-3-319-46478-7_31
30. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Computer Vision and Pattern Recognition (2011)
31. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
32. Zhang, X., Zhang, L., Wang, X.J., Shum, H.Y.: Finding celebrities in billions of web images. IEEE Trans. Multimedia **14**(4), 995–1007 (2012)