Neurocomputing 494 (2022) 443–445

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Original software publication

Face.evoLVe: A cross-platform library for high-performance face analytics

Qingzhong Wang [a,1], Pengfei Zhang [b,1], Haoyi Xiong [a,*], Jian Zhao [c,*]

[a] *Baidu Inc., Beijing, China*
[b] *National University of Defense Technology, Changsha, China*
[c] *Institute of North Electronic Equipment, Beijing, China*

A R T I C L E I N F O

Article history:
Received 20 August 2021
Revised 2 February 2022
Accepted 23 April 2022
Available online 26 April 2022

Keywords:
Face analytics
Toolbox
Library
PyTorch
PaddlePaddle

A B S T R A C T

We develop face.evoLVe—a comprehensive library that collects and implements a wide range of popular deep learning-based methods for face recognition. The motivation of the software is to lower the technical burdens in reproducing the existing methods for comparison, while users of our library could focus on developing advanced approaches more efficiently. More specifically, Face.evoLVe is well designed with an extensible framework under vibrantly evolving, so that new face recognition approaches can be easily plugged into our framework. The library is available at https://github.com/ZhaoJ9014/face.evoLVe. Face.evoLVe has been widely used for face analytics, receiving 2,700 stars and 683 forks and we have used Face.evoLVe to participate in a number of face recognition competitions and secured the first place.

1. Introduction

While deep learning-based face recognition approaches have been reported to outperform human's perception [1], they have been concerned with issues of reproducibility. It might be difficult to achieve the same performance when algorithms and models were re-implemented from the details released in the papers. Although some researchers release codes, it is still inconvenient to reproduce the experiments for fair comparisons as they often have been "cooked with recipes", *e.g.*, tricks in architectures, losses, data pre-processing, and training tricks. Hence, developing a comprehensive face recognition library, with all alternating backbones, loss functions and bag of tricks incorporated, is vital for both researchers and engineers. To this end, we develop a relatively comprehensive deep face recognition library named *face.evoLVe* to meet the goals above. In this paper, we present the features of the developed Face.evoLVe library in details.

We have used Face.evoLVe to participate in a number of face recognition competitions and secured the first place, such the ICCV 2017 MS-Celeb-1 M Large-Scale Face Recognition Hard Set/Random Set/Low-Shot Learning Challenges. Please refer to the online appendix for more results [2].

2. Software framework

2.1. Software architecture

As shown in Fig. 1, the Face.evoLVe library is with three modules: (1) *Data Pre-processors for face detection & alignment*, (2) *Backbone Networks for facial feature extraction*, and (3) *Heads with Loss functions for specific face recognition tasks*. Specifically, with an image as the input data, Face.evoLVe first detects faces in the input and localize the facial landmarks for alignment, where MTCNN [3] is adopted for Pre-processing. The aligned faces are fed into the backbone networks (*e.g.*, ResNet [4], HRNet [5] *etc.*) to extract features for facial representation and recognition, where Face.evoLVe reproduces and implements 13 interchangeable Backbone networks in the library. Finally, 16 alternating head blocks (*e.g.*, Arcface [6], AdaCos [7] *etc.*) have been implemented and included in Face.evoLVe to deliver the face recognition results, where every Head block has been assigned with a specific Loss function (*e.g.*, Focal [8]) for model training. Note that the supports to distributed training are available to accelerate processes.

* Corresponding author.
E-mail addresses: xionghaoyi@baidu.com (H. Xiong), zhaojian90@u.nus.edu (J. Zhao).
[1] Equal contribution

https://doi.org/10.1016/j.neucom.2022.04.118
0925-2312/© 2022 Published by Elsevier B.V.

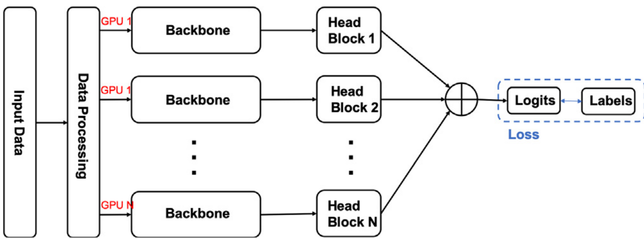

Fig. 1. An overview of the Face.evoLVe framework.

Table 1
Comparison between Face.evoLVe and other libraries. InsightFace [9] is the official implementation of Arcface [6].

Library	# backbones	# heads	Platforms	# forks	# stars
InsightFace [9]	5	4	Pytorch [10], MxNet [11], PaddlePaddle [12]	3,600	11,100
FaceX-zoo [13]	9	8	Pytorch [10]	296	1,200
Face.evoLVe	13	16	Pytorch [10], PaddlePaddle [12]	682	2,700

2.2. Software functionalities

In addition to above functionalities, Face.evoLVe also provides extra options to facilitate model design and training.

(1)Data Balancing and Augmentation. Given an imbalanced data-set, Face.evoLVe removes the low-shot classes (*i.e.*, the faces appear infrequently in the dataset) and incorporates a weighted random sampling to balance the distribution of classes (*i.e.*, faces for recognition). Furthermore, data augmentation strategies, including horizontal flips, hue/saturation/brightness re-scaling, and PCA noises, have been also provided.
(2)Learning Rate Adjustment and Model Tweaking. During the training procedure, Face.evoLVe facilitates the dynamic configurations of learning rates, such as warm-up, learning rate decay, and cosine annealing. Furthermore, it also allows users to reconfigure Backbone networks through changing the strides or resolutions of convolution kernels in certain layers.
(3)Label Smoothing and Knowledge Distillation. In addition to learning from the labels in the form of one-hot vectors, Face. evoLVe provides options to learn from smoothed labels, where the smoothed ones could be obtained from certain distributions (*i.e.*, smoothing one-hot vectors) or the outputs of pre-trained models (*e.g.*, self-training in semi-supervised training with unlabeled face images), so as to boost the performance.
(4)Comparison with Other Libraries. We show the compare Face. evoLVe with two popular libraries – InsightFace [9] and FaceX. Zoo [13] in Table 1. Compared with InsightFace – the official implementation of ArcFace [6], we implement more heads (16 vs. 4) and backbones (13 vs. 5) in face.evoLVe. And compared with FaceX.Zoo, Face.evoLVe contains more backbones (13 vs. 9), heads (16 vs. 8), and support multiple platforms, providing more choices for users. Looking at the popularity (forks and stars) in the community of face analytics, Face.evoLVe is relatively popular.

3. Implementation and empirical results

The Face.evoLVe library is implemented using both Pytorch [10] and PaddlePaddle [12]. The codes are available online at https://github.com/ZhaoJ9014/face.evoLVe. We present the performances of the implemented models in Table 2 and Table 3, where we report the mean accuracy on each dataset. We train the models using MS-Celeb-1 M [14] dataset (Table 2) and Web260M [15] (Table 3), then test them on different datasets. Compared with the state-of-the-art performance, the implemented models in Face.evoLVe achieve competitive results, *e.g.*, the original Arcface [6] model with ResNet100 achieves 99.82 on LFW, 95.45 on CALFW and 92.08 on CPLFW, in contrast, using Face.evoLVe with IR50 achieves 99.78 on LFW, 95.87 on CALFW and 92.45 on CPLFW (see Table 2), performing even better than the original Arcface. Please check out our online appendix [2] for more comparisons and results.

Table 2
The performance of the implemented models in the Face.evoLVe library on different heads and backbones. We train the models using Ms-Celeb-1 M dataset [14].

Backbone	Head	Loss	Testing Dataset						
			LFW	CFP_FF	CFP_FP	AgeDB	CALFW	CPLFW	Vggface2_FPz
IR50	Arcface	Focal	99.78	99.69	98.14	97.53	95.87	92.45	95.22
IR101	Arcface	Focal	99.81	99.74	98.25	97.77	95.93	92.74	95.44
IR152	Arcface	Focal	99.82	99.83	98.37	98.07	96.03	93.05	95.50
IR50	AdaCos	Focal	99.75	99.53	98.39	97.25	95.55	92.25	95.27
IR101	AdaCos	Focal	99.78	99.59	98.41	97.33	95.68	92.41	95.35
IR152	AdaCos	Focal	99.81	99.65	98.42	98.47	95.74	92.57	95.43
IR50	AM-Softmax	Focal	99.59	99.59	98.23	97.37	95.23	92.37	95.35
IR101	AM-Softmax	Focal	99.65	99.62	98.31	97.42	95.38	92.46	95.47
IR152	AM-Softmax	Focal	99.72	99.71	98.45	97.49	95.50	92.52	95.55

Table 3
The performance of the implemented models in the Face.evoLVe library on different heads and backbones. We train the models using Web260M dataset [15].

Backbone	Head	Loss	Testing Dataset						
			LFW	CFP_FF	CFP_FP	AgeDB	CALFW	CPLFW	Vggface2_FP
IR152	AdaCos	Focal	99.82	99.84	98.37	98.07	96.03	93.05	95.50
HRNet	MV-Softmax	Focal	99.82	99.51	98.41	97.88	95.43	88.95	94.70
TF-NAS-A	AM-Softmax	Focal	99.82	99.47	98.33	96.65	94.32	84.88	91.38
GhostNet	ArcFace	Focal	99.69	99.52	98.48	97.29	94.92	85.25	90.88
AttentionNet	AdaCos	Focal	99.82	99.47	98.52	96.89	95.12	87.23	94.23
MobileFaceNet	AdaCos	Focal	99.73	99.84	97.75	95.87	94.87	89.29	93.20

Fig. 2. Face Recognition from a scene of "The Big Bang Theory" TV series.

Table 4
Software metadata.

Nr.	(executable) Software metadata description	Please fill in this column
S1	Current software version	v1.0.
S2	Permanent link to executables of this version	https://github.com/ZhaoJ9014/face.evoLVe.git
S3	MIT License (MIT)	
S4	Computing platform/Operating System	Linux or macOS with Python 3.7 (for training and testing) and Python 2.7 (for visualization)
S5	Installation requirements & dependencies	https://github.com/ZhaoJ9014/face.evoLVe/blob/master/README.md#pre-requisites
S6	If available, link to user manual – if formally published include a reference to the publication in the reference list	https://github.com/ZhaoJ9014/face.evoLVe/blob/master/README.md and an online appendix available at [2]
S7	Support email for questions	zhaojian90@u.nus.edu

4. Illustrative examples

Fig. 2 presents an illustrative example of face detection and recognition from an image. The source codes of this example are available athttps://github.com/ZhaoJ9014/face.evoLVe#face-alignment.

5. Conclusions

In this paper, we have presented face.evoLVe–a comprehensive face recognition library, which offers components covering the full pipeline of face recognition practices, including data pre-processors, backbone networks, and heads with losses. In addition, Face.evoLVe is designed in a highly modular and extensible manner, where users could easily implement and plug their own models into the library for potential extension. Currently, Face.evoLVe is still evolving with a group of active contributors. Commitments with novel models, tricks and datasets are welcome.

Current executable software version

See Table 4.

CRediT authorship contribution statement

Qingzhong Wang: Validation, Formal analysis, Data curation, Visualization, Writing – original draft. **Pengfei Zhang:** Software, Visualization. **Haoyi Xiong:** Supervision, Writing – original draft, Writing – review & editing, Funding acquisition. **Jian Zhao:** Conceptualization, Methodology, Software, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the National Key R&D Program of China (No. 2021ZD0110303), National Science Foundation of China (No. 62006244), and Young Elite Scientist Sponsorship Program of China Association for Science and Technology (No. YESS20200140).

References

[1] Y. Sun, X. Wang, et al., Deep learning face representation from predicting 10,000 classes, CVPR (2014) 1891–1898.
[2] Q. Wang, P. Zhang, et al., Face.evolve: A high-performance face recognition library, arXiv preprint arXiv:2107.08621..
[3] K. Zhang, Z. Zhang, et al., Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Proces. Lett. 23 (10) (2016) 1499–1503.
[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR (2016) 770–778.
[5] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al., Deep high-resolution representation learning for visual recognition, T-PAMI..
[6] J. Deng, J. Guo, et al., Arcface: Additive angular margin loss for deep face recognition, in: CVPR, 2019, pp. 4690–4699..
[7] X. Zhang, R. Zhao, Y. Qiao, X. Wang, H. Li, Adacos: Adaptively scaling cosine logits for effectively learning deep face representations, in: CVPR, 2019, pp. 10823–10832..
[8] T.-Y. Lin, P. Goyal, et al., Focal loss for dense object detection, in: ICCV, 2017, pp. 2980–2988..
[9] J. Guo, J. Deng, X. An, J. Yu, Insightface, https://github.com/deepinsight/insightface.git (2019)..
[10] A. Paszke, S. Gross, et al., Pytorch: An imperative style, high-performance deep learning library, NeurIPS 32 (2019) 8026–8037.
[11] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, Z. Zhang, Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, arXiv preprint arXiv:1512.01274..
[12] Y. Ma, D. Yu, T. Wu, H. Wang, Paddlepaddle: An open-source deep learning platform from industrial practice, Front. Data Comput. 1 (1) (2019) 105–115.
[13] J. Wang, Y. Liu, Y. Hu, H. Shi, T. Mei, Facex-zoo: A pytorh toolbox for face recognition, arXiv preprint arXiv:2101.04407..
[14] Y. Guo, L. Zhang, et al., Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, in: ECCV, 2016, pp. 87–102..
[15] Z. Zhu, G. Huang, et al., Webface260m: A benchmark unveiling the power of million-scale deep face recognition, in: CVPR, 2021..

Qingzhong Wang is currently a Senior R&D Engineer at Big Data Lab, Baidu Research. He received his Ph.D degree in computer science from City University of Hong Kong in 2021. He received the M.Eng and B.Eng from Harbin Engineering University in 2016 and 2013, respectively. His research interests include computer vision and multimodality.

Pengfei Zhang received Ph.D degree in computer science from National University of Defense Technology, Hunan, China. His research interests include computer vison and face recognition.

Haoyi Xiong is currently a Principal R&D Architect at Big Data Lab, Baidu Research. He received his Ph.D degree in Computer Science from Télécom SudParis (now a part of Institut Polytechnique de Paris) jointly with Pierre and Marie Curie University, Paris 6 (now a part of Sorbonne Université), Paris, France in 2015. His research interests include ubiquitous computing and automated deep learning.

Jian Zhao currently an Assistant Professor with Institute of North Electronic Equipment, Beijing, China. He received his Ph.D. degree from National University of Singapore (NUS) in 2019, Master degree from National University of Defense Technology (NUDT) in 2014, and Bachelor degree from Beihang University (BUAA) in 2012. His research interests include computer vision and deep learning.