JOHANNES KEPLER
UNIVERSITY LINZ

Philipp Hofer
Institute of
Networks and Security

@ philipp.hofer@ins.jku.at
https://www.digidow.eu/

March 2021

Analysis of state-of-the-art off-the-shelve face recognition pipelines

Technical Report

Christian Doppler Laboratory for
Private Digital Authentication in the Physical World

This work has been carried out within the scope of Digidow, the Christian Doppler Laboratory for Private Digital Authentication in the Physical World, funded by the Christian Doppler Forschungsgesellschaft, 3 Banken IT GmbH, Kepler Universitätsklinikum GmbH, NXP Semiconductors Austria GmbH, and Österreichische Staatsdruckerei GmbH.

INSTITUTE
OF NETWORKS
AND SECURITY

DIGiDOW

**JOHANNES KEPLER
UNIVERSITY LINZ**
Altenberger Straße 69
4040 Linz, Austria
jku.at

Contents

1. Introduction

Face recognition pipelines are under active development, with many new publications every year. The goal of this report is to give an overview a modern pipeline and recommend a state-of-the-art approach while optimizing for accuracy and performance on low-end hardware, such as a Jetson Nano.

2. Pipeline

Generally, most state-of-the-art face recognition pipelines distinguish between

- **Face detection:** Finding faces in images
- **Face recognition:** Mapping faces to a particular person

In order to be able to create a real-time system, the pipeline should be able to process at least 3 frames every second. Thus, face detection and recognition should take at most 300 ms.

3. State-of-the-art face detection models

In this report we analyze the three most popular current state-of-the-art face detection models Retinaface [2], MTCNN [6], and Faceboxes [7].

In order to objectively compare their accuracy, we evaluated the networks on the evaluation set of the WIDER Easy Face dataset [5]:

- Retinaface: 94.21%
- MTCNN: 91%
- Faceboxes: 86.3%

Next, we evaluated their speed on a 1080p image, executed on an *Intel Core i5-8265U CPU*:

- Retinaface: 750 ms (1.3 FPS)
- MTCNN: 550 ms (1.8 FPS)
- Faceboxes: 35 ms (28 FPS)

4. Results face detection

The trade-off between speed and performance between the various networks is clearly visible. The significant speed improvement of Faceboxes comes with a caveat though: The network only works well on high-quality images, i.e. a large face looking straight into the camera. Furthermore, both Retinaface and MTCNN also calculate five landmarks of the face, which are needed for face recognition.

For further research, there are many opportunities for tweaking certain elements in order to increase speed:

	ArcFace	FaceNet
Laptop (Intel Core i5-8265U CPU)	0.21 s / face embedding	0.17 s / face embedding
Pi 3	3.5 s / face embedding	3.1 s / face embedding
Pi 4	2 s / face embedding	1.5 s / face embedding

Table 1: Speed comparison of 2 state-of-the-art face-recognition algorithms.

- Use a different (smaller) model. Face detection networks use *backbone networks* for extracting features from images. The size of the network (mainly network depth) plays a significant role with respect to the inference time. Therefore, one option to achieve faster inference time (at the expense of accuracy) is to move to a smaller backbone model.
- Reduce the image size. Downscaling an image would result in faster inference time. This is probably not useful in practice, as you simply waste camera-quality.
- Run face detection only on parts of the whole images, e.g. only where some changes happened or where a face has previously been detected.
- Use a simpler model to generate *face-proposals*. Crop the image to these proposals and run the full model on these proposals only.

5. Face recognition

In order to be able to quickly compare a face to many (i.e. millions) other faces, an embedding is extracted. Most papers which have been published in the last years (e.g. ArcFace [1], SphereFace [3], CosFace [4]) use an embedding size of 512.

We compared the speed of two state-of-the-art face recognition algorithms – ArcFace and Facenet. The results are shown in Table 1.

Using the GPU (either on the laptop or on an embedded device, such as a Jetson Nano) would increase speed performance.

References

[1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4690–4699.

[2] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. Retinaface: single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*.

[3] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Sphereface: deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

[4] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5265–5274.

[5] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. 2016. Wider face: a face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5525–5533.

[6] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23, 10, 1499–1503.

[7] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. 2017. Faceboxes: a cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 1–9.