International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Factors Affecting Deployment of Deep Learning based Face Recognition on Smartphones

Abhijith Kuriakose[1], Atharva Kadam[2], Prajakta Gaikwad[3], Saloni Sambherao[4], Anagha Chaudhari[5]

[1-5]*Pimpri Chinchwad College of Engineering, Nigdi, Pune - 411044, India*

Abstract - *Face recognition is being increasing used on smartphones for user authentication, with more recent technologies (such as Apple's TrueDepth camera system) giving better results than even fingerprint authentication. Traditional methods based on hand engineered features (such as edges and texture descriptors), combined with machine learning techniques have been replaced by deep learning methods based on CNN due to significant accuracy improvement. In this paper, we examine the factors affecting the deployment of deep learning models for face recognition on smartphones.*

Key Words: Face Recognition, Datasets, Knowledge Distillation, Loss Function

1.INTRODUCTION

According to a 2018 survey by gsmarena [1], on average, new smartphones on the market have 14.5MP cameras, 4.3GB RAM and score 2379 on the Basemark OS II benchmark.

A typical smartphone camera uses a 4:3 aspect ratio and has a horizontal FOV between 60° and 80°. The average width of a human face is 16cm and 100x100 pixel faces are typically used for face recognition. Under these conditions, the maximum distance a smartphone can be used for face recognition would be 7.89m. By working with lower resolution images (up to 32x32 pixels), we can increase the distance further (up to 24.67m). Readers are referred to Pei *et al.* [2] for a comprehensive review on approaches to Low Quality Face Recognition (LQFR).

A typical face recognition system has the following stages:
• *Face Detection* is used to detect faces in the images. With the rise in GPU availability and better training sets, deep convoluted neural networks (DCNN) are been increasingly used for this task. The speed of face detectors is still a crucial bottleneck in the face recognition pipeline. SSD and YOLO provide a fast solution for face detection. DPSS is a multi-scale face detector that can produce reliable and accurate face detections at different scales, thus making it capable of detecting tiny and blurred faces.

• *Face normalization* provides an effective and cheap way to distill face identity and dispel face variances, making face recognition easier. This includes steps such as changing orientation, lighting normalization, etc. Zheng *et al.* [6] examines traditional approaches for face normalization. A newer approach using a Generative Adversarial Network (GAN) is presented by Qian *et al.* [7].

• At the *Feature Extraction* stage, the pixel values of a face image are transformed into a compact and discriminative feature vector, also known as a template. Ideally, all faces of the same subject should map to similar feature vectors.

• In the *Feature Matching* building block, two templates are compared to produce a similarity score that indicates the likelihood that they belong to the same subject.

The last two stages are collectively called *Face Recognition*. Traditional methods based on hand engineered features (such as edges and texture descriptors), combined with machine learning techniques (such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) or Support Vector Machines (SVM)) have been replaced by deep learning methods based on CNN (automatically determines the best features to represent a face) due to significant accuracy improvement.

The factors to consider while training a CNN are datasets, architecture and loss function. While deploying to mobile devices, we need to be mindful about resource constraints. This can be achieved by using model compression techniques. In the next section, we explore these factors and techniques.

2. Literature Review

2.1 Training Data

In general, CNNs trained for classification become more accurate as the number of samples per class increases. This is because the CNN model can learn more robust features when is exposed to more intra-class variations. However, in face recognition we are interested in extracting features that generalize to subjects not present in the training set. Hence, the datasets used for face recognition must also contain many subjects so that the model is exposed to more inter-class variations.

The effect that the number of subjects in a dataset was studied in [11]. In this work, a large dataset was first sorted

by the number of images per subject in decreasing order. Then, a CNN was trained with different subsets of training data by gradually increasing the number of subjects. The best accuracy was obtained when the first 10,000 subjects with the most images were used for training. Adding more subjects decreased the accuracy since too few images were available for each extra subject.

Another study [12] investigated whether wider datasets are better than deeper datasets or vice versa (a dataset is considered deeper than another if it contains more images per subject and is considered wider than another if it contains more subjects). From this study, it was concluded that given the same number of images, wider datasets provide better accuracy. The authors reason that this is because wider datasets contain more inter-class variations and, therefore, generalize better to unseen subjects.

Some common public datasets used for face recognition are:

Table -1: Comparison of Datasets

Comparison of Datasets			
Dataset	**Images**	**Subject**	**Images per Subject**
CelebA [13]	202,599	10,177	19.9
VGGFace [15]	2.6M	2,622	1,000
VGGFace2 [16]	3.31M	9,131	362.6

2.2 Architecture

The Facebook's DeepFace, one of the first CNN-based approaches for face recognition that used a high capacity model, achieved an accuracy of 97.35% on the LFW benchmark, reducing the error of the previous state-of-the-art by 27%. The authors trained the CNN with softmax loss using a dataset containing 4.4 million faces from 4,030 subjects.

Two novel contributions were made in this work:

• An effective facial alignment system based upon explicit 3D modelling of faces

• A CNN architecture containing locally connected layers that (unlike regular convolutional layers) can learn different features from each region in an image.

Recent work shows that CNNs can be substantially deeper, more efficient and accurate to train if they contain shorter connections between layers close to the input and those close to the output.

ResNets [21] have become the preferred choice for many object recognition tasks, including face recognition. The main novelty of ResNets is the introduction of a building block that uses a shortcut connection to learn a residual mapping. The use of shortcut connections allows training of much deeper architectures as they facilitate the flow of information across layers. A study of different CNN architectures was carried out in [17]. The best trade-off between accuracy, speed and model size was obtained with a 100-layer ResNet with a residual block like the one proposed in [18].

DenseNet [23] follows a principle similar to ResNet, except that it connects all previous layers (in a dense block) and not just the previous two/three layers.

MobileNets [22] are being developed and used for real-time face recognition on devices with limited computational resources.

2.3 Loss Function

The choice of loss function for training CNN-based methods has been the most recent active area of research in face recognition. Even though CNNs trained with *softmax* loss have been very successful, it has been argued that the use of this loss function does not generalize well to subjects not present in the training set. This is because the softmax loss is encouraged to learn features that increase inter-class differences (to be able to separate the classes in the training set) but does not necessarily reduce intra-class variations. Several methods have been proposed to mitigate this issue. A simple approach is to optimize the bottleneck features using a discriminative subspace method such as joint Bayesian.
Another approach is to use metric learning. *Triplet loss* function [14] is one of the most popular metric learning approaches for face recognition. The aim of the triplet loss is to separate the distance between positive pairs from the distance between negative pairs by a margin. In practice, CNNs trained with triplet loss converge slower than with softmax loss due to the large number of triplets (or pairs in the case of contrastive loss) needed to cover the entire training set. This problem can be mitigated by selecting hard triplets (i.e. triplets that violate the margin condition) during training.

An alternative loss function used to learn discriminative features is the *center loss* proposed in [19]. The goal of the center loss is to minimize the distances between bottleneck features and their corresponding class centers. By jointly training with softmax and center loss, it was shown that the features learnt by a CNN could effectively increase inter-personal variations (softmax loss) and reduce intra-personal variations (center loss). The center loss has the advantage of being more efficient and easier to implement than the contrastive and triplet losses since it does not require forming pairs or triplets during training.

Another related metric learning method is the *range loss* proposed in [20] for improving training with unbalanced datasets. The range loss has two components. The intra-class component of the loss minimizes the k-largest distances between samples of the same class, and the inter-class component of the loss maximizes the distance between the closest two class centers in each training batch. By using these extreme cases, the same information from each class is used, regardless of how many samples per class are available.

Range loss needs to be combined with softmax loss to avoid the loss being degraded to zeros [19].

Rajan *et al.* [4] recently introduced *crystal loss* which functions by constraining the deep features to lie on a hypersphere and achieves impressive results on the IARPA Janus Benchmark C (IJB-C) dataset.

2.4 Model Compression

Training must extract structure from highly redundant, very large datasets but it does not need to operate in real time, and it can use a huge amount of computation. Deployment to users, however, has much more stringent requirements on latency and computational resources. Hence, we need to make use of model compression techniques such as pruning and factorization.

In [5], Hilton *et al.* suggest that a smaller model can be trained by using the results from the original model as targets and called this process "distillation". A distilled network was shown to have comparable accuracy to the original in [8].

2.5 Generative Adversarial Network (GAN)

In a GAN [10], two neural networks (generator and discriminator) are pitted against each other. In the case of identifying real faces, the generators goal is to take random values and create fake faces that the discriminator thinks are real, while the discriminators goal is to successfully identify real and fake faces.

Advances in GAN allows us to generate additional training images without requiring millions of face images to be labelled. Recent works [9] allow facial attribute manipulation, facial expression editing, face frontalisation and face ageing. We can also generate novel identities using GANs [3].

3. CONCLUSIONS

By using CNN for face recognition, we get the following:

Advantages:
- Significant accuracy improvement in comparison to traditional methods based on hand-crafted features.
- Straightforward system scaling to achieve even higher accuracy by increasing the size of the training sets and/or the capacity of the networks.

Disadvantages:
- Needs to be trained with very large datasets of labelled face images that contain enough variations to generalize unseen samples. Collecting such datasets is expensive.
- Very deep CNN architectures are slow to train.
- Very deep CNN architectures are hard to deploy.

Proposed Solutions:
- GANs are a promising solution to the first issue of data collection. They can be used to generate additional training images without requiring millions of face images to be labelled.
- To address the second issue of training time, more efficient architectures such as MobileNets can be used
- The deployment issue can be dealt with by using model compression techniques.

REFERENCES

[1] Paul, "Smartphones in 2018: the half year report", GSMarena, July 2018
[2] Pei Li, Patrick J. Flynn, Loreto Prieto, Domingo Mery, "Face Recognition in Low Quality Images: A Survey", arXiv, March 2019
[3] Chris Donahue, Zachary C. Lipton, Akshay Balsubramani, Julian McAuley, "Semantically Decomposing the Latent Spaces of Generative Adversarial Networks", arXiv, February 2018
[4] Rajeev Ranjan, Ankan Bansal, Hongyu Xu, Swami Sankaranarayanan, Jun-Cheng Chen, Carlos D Castillo, Rama Chellappa, "Crystal Loss and Quality Pooling for Unconstrained Face Verification and Recognition", arXiv, February 2019
[5] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the Knowledge in a Neural Network", arXiv, March 2015
[6] Sheetal Tulshidas Chaudhari, Archana Kale, "Face Normalization: Enhancing Face Recognition", IEEE, November 2010
[7] Yichen Qian, Weihong Deng, Jiani Hu, "Unsupervised Face Normalization with Extreme Pose and Expression in the Wild", CVPR, 2019
[8] Francesco Guzzi, Luca De Bortoli, Stefano Marsi, Sergio Carrato, Giovanni Ramponi, "Distillation of a CNN for a high accuracy mobile face recognition system", IEEE, July 2019
[9] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, Xilin Chen, "AttGAN: Facial Attribute Editing by Only Changing What You Want", arXiv, July 2018
[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Nets", arXiv, June 2014
[11] Erjin Zhou, Zhimin Cao, Qi Yin, "Naive-Deep Face Recognition: Touching the Limit of LFW Benchmark or Not?", arXiv, January 2015
[12] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, Rama Chellappa, "The Do's and Don'ts for CNN-based Face Verification", arXiv, September 2017
[13] S. Yang, P. Luo, C. C. Loy, X. Tang, "From Facial Parts Responses to Face Detection: A Deep Learning Approach", IEEE, 2015
[14] Kilian Q. Weinberger, Lawrence K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification", JMLR, February 2009
[15] O. M. Parkhi, A. Vedaldi, A. Zisserman, "Deep Face Recognition", British Machine Vision Conference, 2015
[16] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, "VGGFace2: A dataset for recognising face across pose and age", International Conference on Automatic Face and Gesture Recognition, 2018

[17] Jiankang Deng, Jia Guo, Niannan Xue, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", arXiv, February 2019
[18] Yoshihiro Yamada, Masakazu Iwamura, Koichi Kise, "Deep Pyramidal Residual Networks with Separated Stochastic Depth", arXiv, December 2016
[19] Yandong Wen, Kaipeng Zhang, Zhifeng Li, Yu Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition", Springer, 2016
[20] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, Yu Qiao, "Range Loss for Deep Face Recognition with Long-tail", arXiv, November 2016
[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", arXiv, December 2015
[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", arXiv, March 2019
[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, "Densely Connected Convolutional Networks", arXiv, January 2018