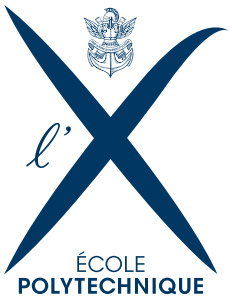


Jeremy Gozlan
Martin Cepeda

December 17, 2020
Intermediary report

HIGH RESOLUTION FACE RECOGNITION

M2 Transverse project with IDEMIA

Contents

1 Introduction

Face recognition is a known problem that still *faces* (not pun intended) a lot of challenges when the number of possible identities is to be extremely large. Current state of the art techniques focus on the design of sophisticated loss function that enhance discriminative power of learnt features. Among such losses, angular-based losses such as ArcFace (Deng et al. [2019]) and CosFace (Wang et al. [2018]) include margins in order to maximise class separability and thus provide lesser ambiguity when doing classification.

Within the applications of face recognition algorithms we can cite identification and identity safety, two areas in which IDEMIA is arguably the world leader in the subject, with a presence in 180 countries, #1 in police biometric systems, civil identity solutions and U.S. driver's license issuance (IDEMIA [2020]). In order to continuously improve the performance of face recognition solutions, one area of research is the possibility to improve high resolution face recognition through the use of a large face dataset of small resolution.

1.1 Objective

This project's goal is to understand the possible benefits and limitations of improving high resolution face recognition through the use of a large face dataset of small resolution.

2 Dataset

For this project, IDEMIA has provided a face identity dataset consisting of:

```
/data
├── highres
│   └── ...
├── highres_eval
│   └── ...
├── hres_eval_pairs
│   └── hres
│       ├── data
│       └── meta
└── lowres
    └── ...
```

Dataset directory tree

- `highres` contains 10.000 faces distributed across 2.500 identities (4 images per class). Each image has a resolution of 144x144 (RGB) and inter-eyes distance of 48 pixels.
- `highres_eval` contains 6.000 faces distributed across 2.000 identities not present in `highres`, 3 images per class). Each image has a resolution of 36x36 (RGB) and inter-eyes distance of 12 pixels.
- `highres_eval_pairs` contains 12.000 faces: 3.000 matching pairs and 3.000 non-matching pairs randomly sampled from `highres`.
- `lowres` contains 1.000.000 faces distributed across 50.000 identities (20 images per class). Each image has a resolution of 36x36 (RGB) and inter-eyes distance of 12 pixels.

The pointers to image files are stored in the files `highres_ids.txt`, `lowres_ids.txt` and `highres_eval_ids.txt`. It is worth to be noted that the dataset is of considerable size (5 GB), that all three folders of images do not contain any identical identities and that transferring the uncompressed files is extremely slow due to the quantity of files.

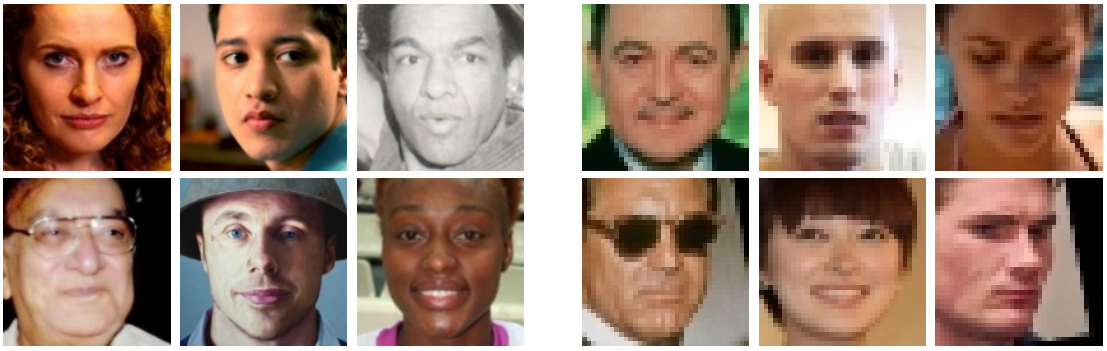


Figure 1: Sample high resolution faces (left) and low resolution faces (right)

Figure 2: Sample validation (high resolution) faces on two different identities

3 Proposed approach

In order to study the feasibility, benefits and limitations of improving high resolution face recognition, we can divide the face recognition pipeline in three main parts, namely 1) Backbone 2) Loss and 3) Training. These three components are available to experimenting under the repository cavaface.pytorch (Li and Chi [2020]) proposed by IDEMIA, which implements several face recognition architectures using PyTorch. In the following subsections we will described the chosen pipeline, which achieves the closest performances to SOTA in face recognition.

3.1 Squeeze-and-Excitation Networks

Introduced by Hu et al. [2019], they are an extension to traditional residual architectures (such as ResNet) that introduce a novel architectural unit, the "Squeeze-and-Excitation" (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels:

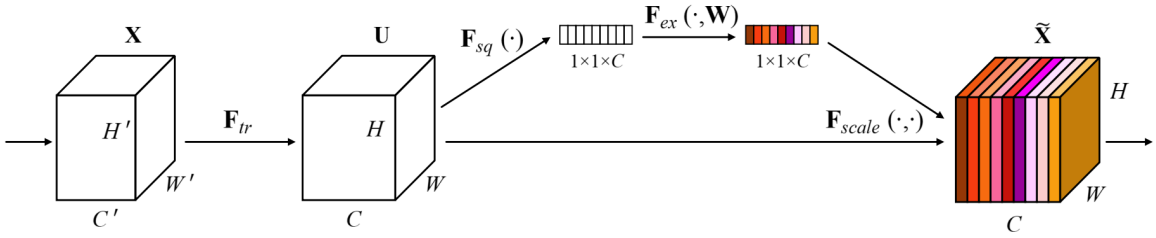


Figure 3: SE block: For any given transformation $\mathbf{F}_{tr}$ mapping the input $\mathbf{X}$ to the feature maps $\mathbf{U}$ where $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ (e.g. a convolution), we can construct a corresponding SE block to perform feature recalibration. The features $\mathbf{U}$ are first passed through a squeeze operation, which produces a channel descriptor by aggregating feature maps across their spatial dimensions $(H \times W)$

The function of this descriptor is to produce an embedding of the global distribution of channel-wise feature responses, allowing information from the global receptive field of the network to be used by all its layers. The aggregation is followed by an excitation operation, which takes the form of a simple self-gating mechanism that takes the embedding as input and produces a collection of per-channel modulation weights. These weights are applied to the feature maps $\mathbf{U}$ to generate the output of the SE block which can be fed directly into subsequent layers of the network.

The *squeeze* operation uses global average pooling to generate channel-wise statistics, and the *excitation* operation employs a simple gating mechanism with a sigmoid activation. With this, it is possible to build IR–SE networks (ResNets with SE blocks) that achieve better performances in image classification and allow for adequately model channel-wise feature dependencies, which is of particular interest for face recognition.

3.2 Additive Angular Margin Loss

Abbreviated ArcFace, this loss was introduced by Deng et al. [2019]. As the dot product between the DCNN feature and the last fully connected layer is equal to the cosine distance after feature and weight normalisation, we can utilise the arc-cosine function to calculate the angle between the current feature and the target weight. This result is then added to an additive angular margin to the target angle to get the target logit back again by the cosine function. Then, a re-scaling of all logits by a fixed feature norm allows to a direct plugging-in of a Softmax loss:

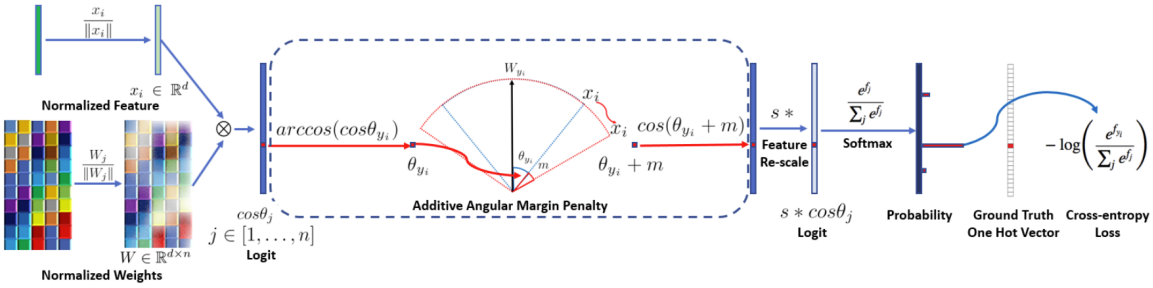
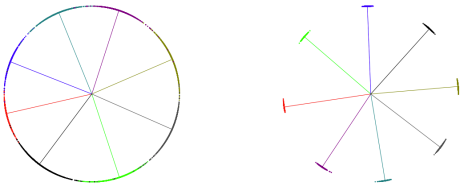


Figure 4: Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $\arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i} . After that, we calculate $\cos(\theta_{y_i} + m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

This loss allows for highly discriminative features for face recognition and improved training as the proposed additive angular margin has a better geometric attribute as the angular margin has the exact correspondence to the geodesic distance margin penalty in the normalised hypersphere:



(a) Softmax (b) ArcFace

Toy examples under the softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature normalisation, all face features are pushed to the arc space with a fixed radius. The geodesic distance gap between closest classes becomes evident as the additive angular margin penalty is incorporated.

3.3 High volume–Low resolution training

The last aspect of a face recognition pipeline concerns precisely how to transfer the information of a large and dimensionally reduced dataset (at the expense of having less information per face) to a heavy but rich high resolution dataset. A common ground on all possible approaches is the up-sampling of low resolution images, which will ultimately allow to apply the trained architecture on both datasets. In order to do this, several leads can be considered:

1. Vainilla transfer learning by training in low resolution and fine-tuning with high resolution images.

2. 2 networks: a pre-trained architecture on low resolution is frozen and provides features from high resolution images in parallel with a simpler network during high resolution training. Features from both networks are concatenated and fed to a single classifier.

3. 2 networks with resolution awareness: same as above but the pre-trained network accepts only low resolution images (high resolution images are down-sampled for the first network and kept for the second, simpler network)

4 Problems and intermediary results

4.1 IR–SE and ArcFace repositories

When starting this project, IDEMIA has been suggesting us to use the cavaface.pytorch implementation as our baseline code. It is high-performance distribute parallel training framework for face recognition that includes a large number of backbone networks and losses including IR–SE and ArcFace. At the beginning of this project, the library was suffering from a lack of updates, old dependencies and presented many errors. Also, being entirely focused on parallel GPU training, the code could not be run on our machines and so had to be entirely adjusted to be run on CPUs. While successful, the training time was so large and the computational requirements so big that we decided to found a way to run it on GPUs.

In the meantime, we also tried a second implementation of ArcFace in a TensorFlow based repository: InsightFace_TF (Wei [2018]). This solution was focusing less on parallelization and CPU adjustments were manageable. However, in late November, cavaface.pytorch was updated which solved many problems and because of TensorFlow lack of computer vision tools, we switched to this newer version. As before, a non parallel CPU/GPU version has been made and tried to be run on cloud GPUs.

4.2 Problems encountered

This project while clearforward in its direction is suffering from major problem, our computational limitations. Unlike small networks with low dimension inputs that can be run on CPUs, it requires an access to a powerful GPU cluster. Indeed, with one million images and a ResNet architecture, the computer we currently have cannot train such models. Most of the work we have been able to done so far was in the direction of solving this issue. We tested the following solutions:

- Google Collab: Due to several abuses on GPUs, Google decided to restrict and limit the access to their GPUs. Which means after a few hours of training (less than one epoch), the instance is stopped. Secondly, transferring such a large number of images to Google Collab is extremely tedious is not stable.

- Qarnot: With Qarnot, the main problem was the only ability to access CPUs. Additionally, data transfer to Qarnot is more than unstable resulting in constant upload errors. We decided to contact the support multiple times first without much success and finally asked if GPUs could be allocated for this project. The answer we got was the fact that only a few GPUs were available and already overbooked and that the credits we got will only last a few hours. We decided to stop trying to get a Qarnot training running as CPUs were not a viable option.

- Microsoft Azure: Because of our student status, it seems that we can have access to 100 dollars of Azure credits per Polytechnique account. We are now investigating this option that seems possible but require a large volume of adaptability work because of the cavaface library multiple dependencies.
- IDEMIA: IDEMIA team has been extremely comprehensible with this problem but because of confidentiality reasons and the fact that we were not employees, they could not justify an access to their computational power. The option we have is to deliver them a script that they could run for us but this option is to the mercy of errors and a total lack of control on the project. However if Azure does not work in the soon future, this will be the only option we have left and the one we will take.

So far, we have been trying to get a model training on a large number of GPUs possible platform without real success due to a lack of credit and options. While a smaller dataset could be use (we have been building one), the ability of small images to give more predictive power to large images require the full dataset to be huge as preliminary IDEMIA work showed that the difference to be noted is expected to be extremely small.

5 Future work

When a baseline large resolution images model will have been trained, our work will mainly focus on the building of the architecture and options suggested.

References

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition, 2019.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

IDEMIA. Corporate brochure. https://www.idemia.com/sites/corporate/files/about-us/download/idemia-corporate-brochure-2020_0.pdf, Jun 2020.

Yaobin Li and Liying Chi. cavaface.pytorch: A Pytorch Training Framework for Deep Face Recognition, 2020. URL https://github.com/cavalleria/cavaface.pytorch.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large Margin Cosine Loss for Deep Face Recognition, 2018.

Chen Wei. Insight face in tensorflow, 2018. URL https://github.com/auroua/InsightFace_TF.