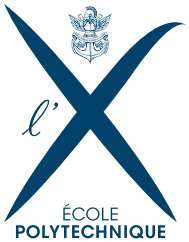
High Resolution Face Recognition

Transverse Project MSc&T AI-ViC2

Martín Cepeda & Jeremy Gozlan

March 18, 2021



INSTITUT
POLYTECHNIQUE
DE PARIS

Outline

1. Introduction
2. SOTA in HSFR
   - `cavaface.pytorch`
   - Angular loss
3. Proposed approach
   - Dataset
   - High volume-Low resolution training
   - Architecture
   - Training and Validation
4. Results
   - Baseline models
   - Fine-tuning
   - Embedding analysis
   - Weight analysis
5. Conclusion & extensions

Introduction

- High scale face recognition (HRFR) is still an open and difficult problem
- Current SOTA techniques focuses on the design of sophisticated losses that enhance discriminative power
- Main drawback: data availability and cost of training in HD

Project context:

- IDEMIA is one of the world leaders in identification and identity safety
- Goal: Understand possible benefits/limitations of improving low resolution face recognition using a small face dataset of high resolution

Outline

1. Introduction
2. SOTA in HSFR
   - `cavaface.pytorch`
   - Angular loss
3. Proposed approach
   - Dataset
   - High volume-Low resolution training
   - Architecture
   - Training and Validation
4. Results
   - Baseline models
   - Fine-tuning
   - Embedding analysis
   - Weight analysis
5. Conclusion & extensions

cavaface.pytorch

HSFR can usually be divided in three parts: Backbone, Loss and Training Procedure.

Multiple libraries providing these blocks exist including cavaface.pytorch, which provides:

- High performance distribute parallel training performance for HSFR in PyTorch
- Various CNN/image specialized network architectures (Backbones)
- Various discriminative losses that can be used for HSFR when a large number of classes exist (Loss)
- Image and Network training techniques like data augmentation or LR warmup (Training)

ArcFace

Softmax is usually used. Does not optimize embedding features to enforce higher similarity for intra-class and diversity for inter-class.

Additive Angular Margin Loss (ArcFace) is a loss designed for deep face recognition:

- Learns very discriminative features giving a linear angular margin
- Achieved SOTA performances on ten FR benchmarks.
- Simple to implement, easily converging and efficient (negligible computational complexity).

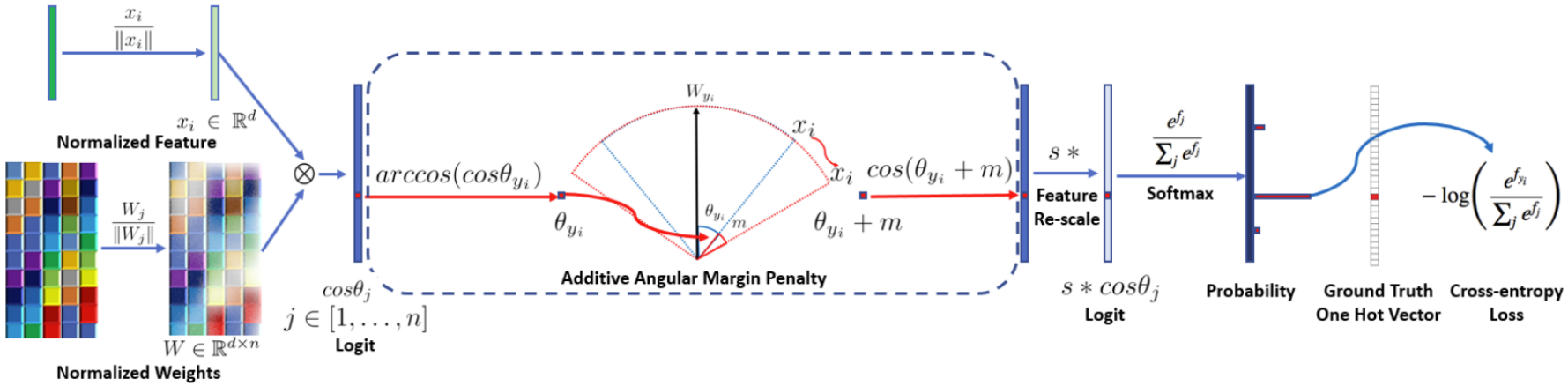


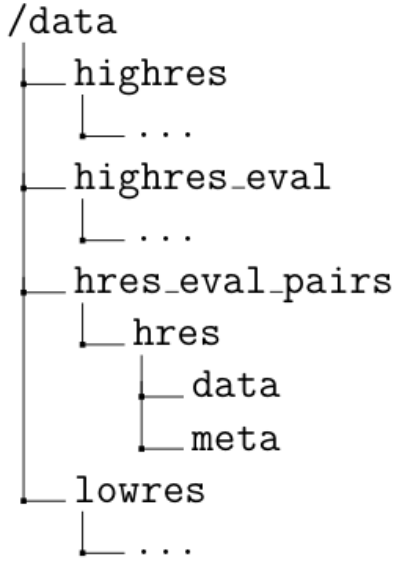
Figure 1: ArcFace schematic description (from [1])

Outline

1. Introduction
2. SOTA in HSFR
   - `cavaface.pytorch`
   - Angular loss
3. Proposed approach
   - Dataset
   - High volume-Low resolution training
   - Architecture
   - Training and Validation
4. Results
   - Baseline models
   - Fine-tuning
   - Embedding analysis
   - Weight analysis
5. Conclusion & extensions

Dataset

For this project, IDEMIA provided us a face identity dataset:



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.

Size: 5 Gb for low resolution images.
No overlapping identities between folders.

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

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

High volume-Low resolution training

Goal: Transfer information from

(large and low resolution dataset) $\rightarrow$ (small but high resolution dataset)

We can upscale low resolution images and train a network on both datasets. How?

- Vanilla transfer learning of a low resolution trained model with high resolution fine-tuning.
- f(low resolution, high resolution) separate networks $\rightarrow$ last layers and head fine-tuning with high resolution images.
- f(low resolution, up-scaled low resolution) separate networks $\rightarrow$ last layers and head fine-tuning with high resolution images.
- Many other variants

Architecture

HSFR pipeline: **Backbone** (embedding network) and **Head** (loss used for classification)

- Backbone: Usually a CNN type, Cavaface options include Resnet, Resnet IRSE, DenseNet and others
- Head: Included in Cavaface are ArcFace, CosPhere, spherical losses and many others

For this project, due to computational limitations, we used a custom Backbone:

- Input layer: Conv2d + Batchnorm + ReLU
- Body: 8 Resnet IRSE blocks (4 distincts used twice).
- Output layer: Batchnorm + Linear (transforms IRSE blocks output – features maps – to a vector

Training and Validation

- Training at first was difficult as the number of low resolution images was too high (Qarnot ✗, Google Colab ✗, Microsoft Azure ✗)
- We manage to train networks through IDEMIA help which offered us to run our scripts
- For validation, we used the untouched evaluation high resolution pair images (6000 pairs, half matching)
- High resolution or Low resolution up-scaled: Image size (144,144,3)
- Low resolution or High resolution down-scaled: Image size (32,32,3)
- Model comparison: Training loss, validation accuracy during training and last epoch ROC curve

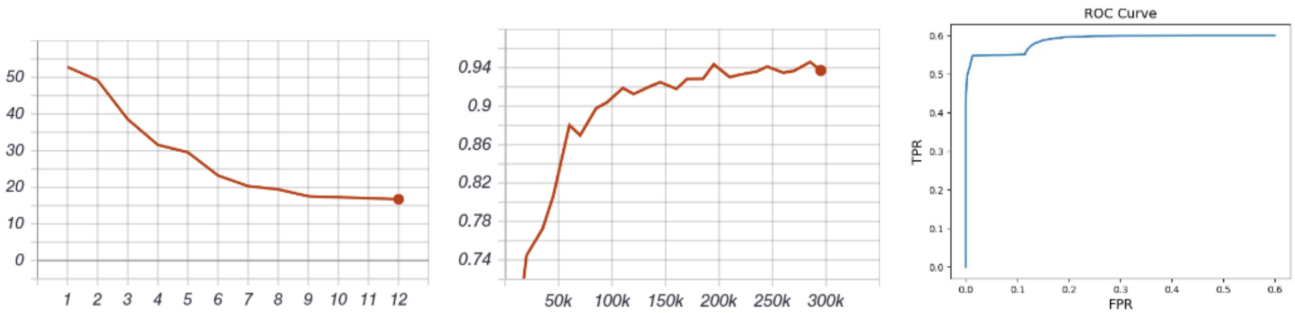
Outline

1. Introduction
2. SOTA in HSFR
   - `cavaface.pytorch`
   - Angular loss
3. Proposed approach
   - Dataset
   - High volume-Low resolution training
   - Architecture
   - Training and Validation
4. Results
   - Baseline models
   - Fine-tuning
   - Embedding analysis
   - Weight analysis
5. Conclusion & extensions

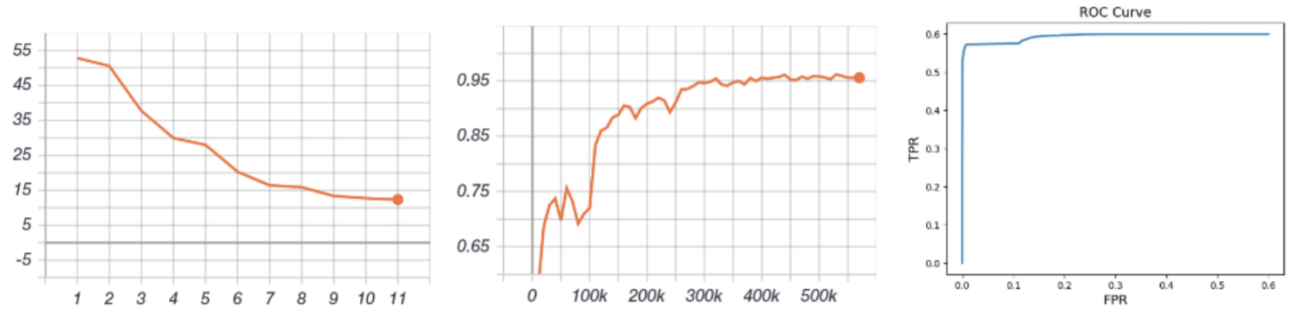
Results: Baseline models

- Left to right: Training loss, Validation accuracy and validation ROC curve
- LR: Low resolution , HR: High resolution

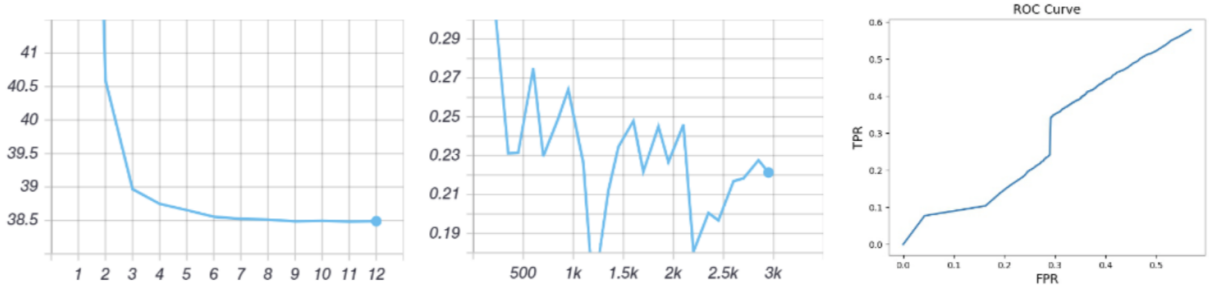
Lowres32: Trained on LR images (1M)



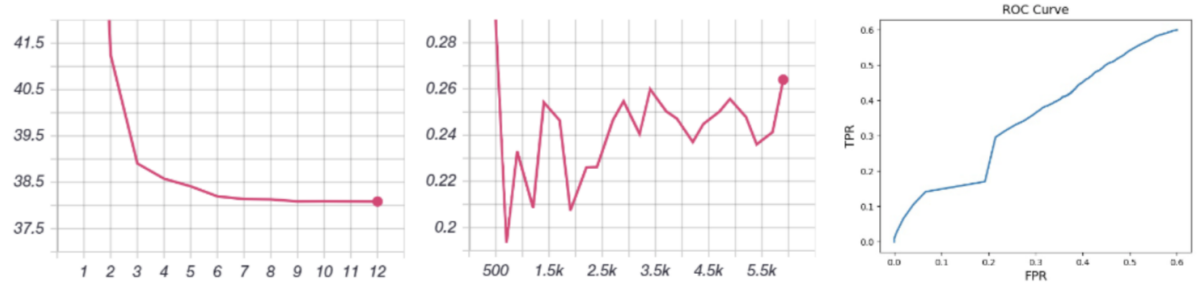
Lowres144: Trained on upscaled LR images (1M)



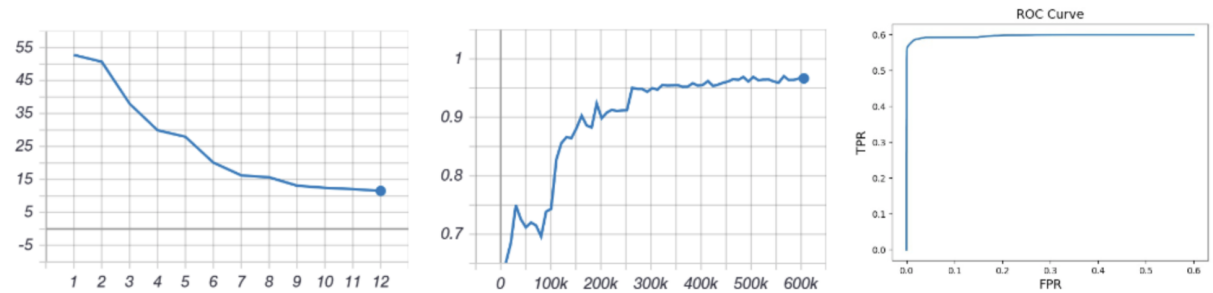
Highres32: Trained on downscaled HR images (10k)



Highres144: Trained on HR images (10k)

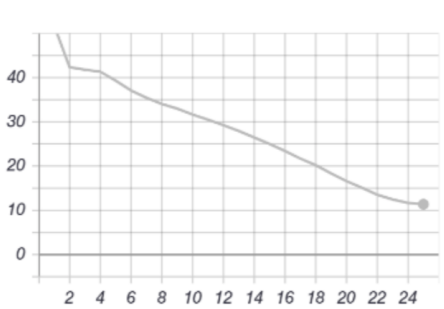


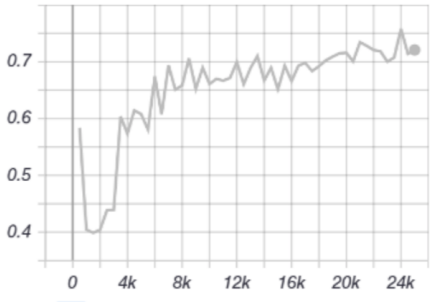
Highlow144: Upscaled LR images + HR images (1M + 10K)

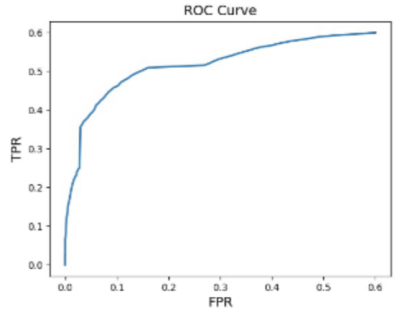


Results: Baseline fine-tuning

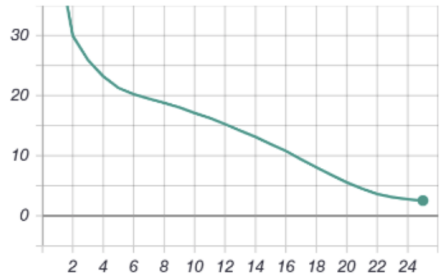
Lowres144 baseline + HR fine-tuning

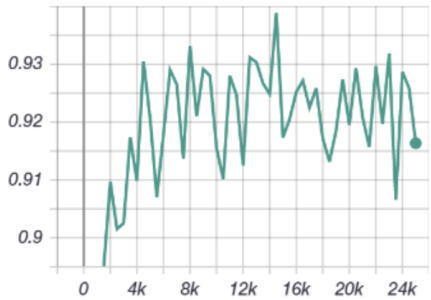


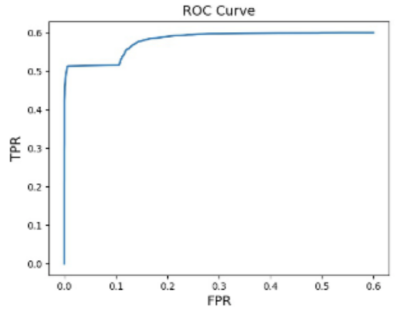




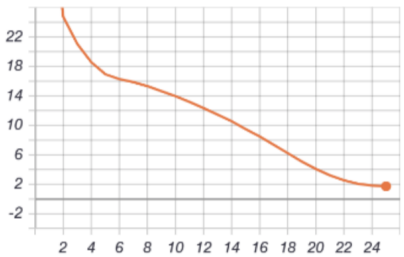
(Lowres144 + Lowres32) + HR fine-tuning for head

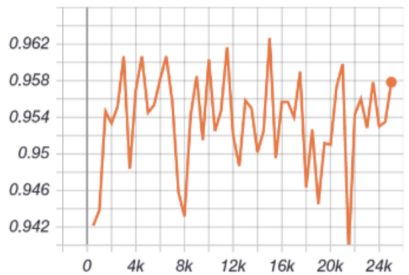


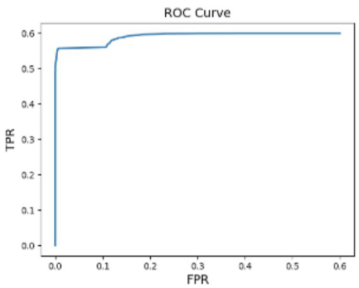




(Lowres144 + Lowres32) + HR finetuning for Backbone output layer

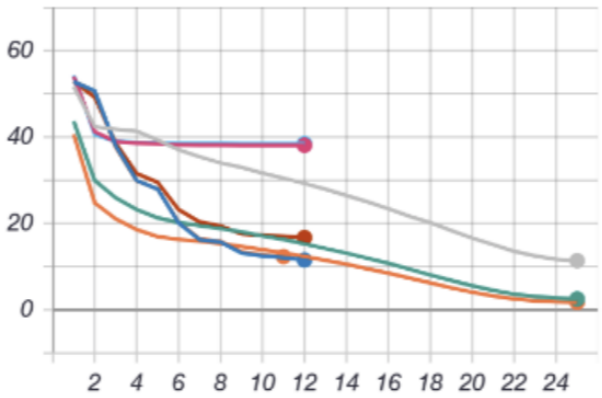


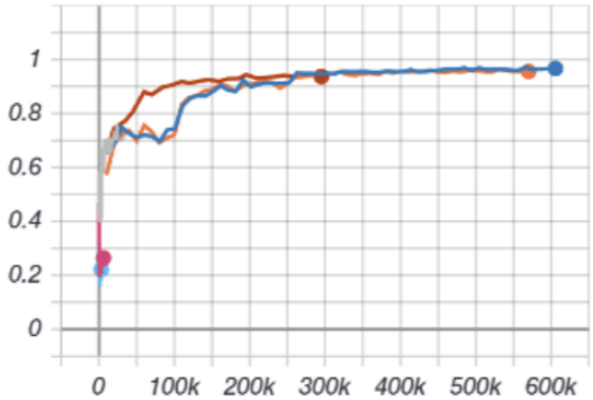


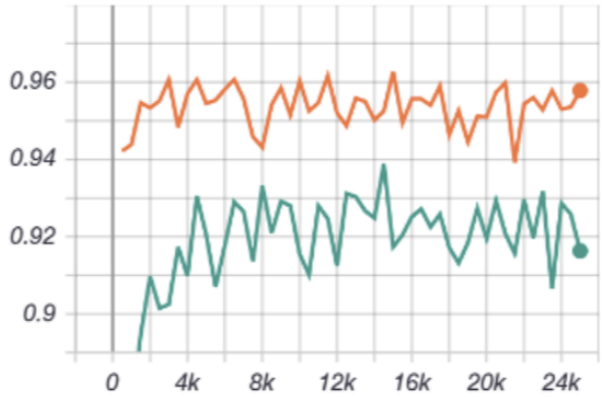


Model Comparison

All models







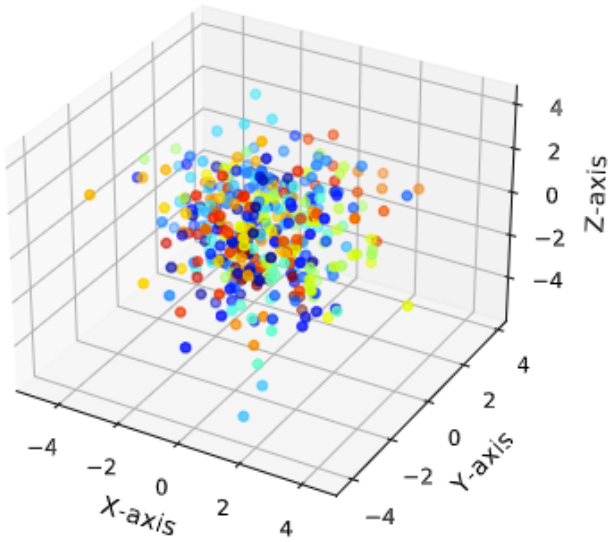
Discussion

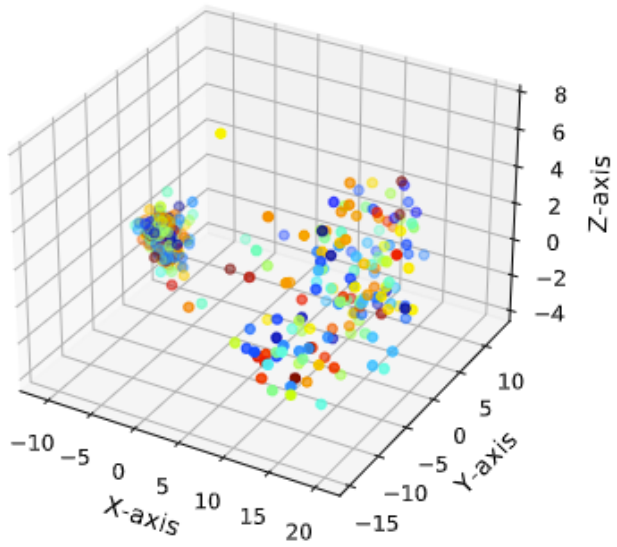
We observed the following interesting behaviours:

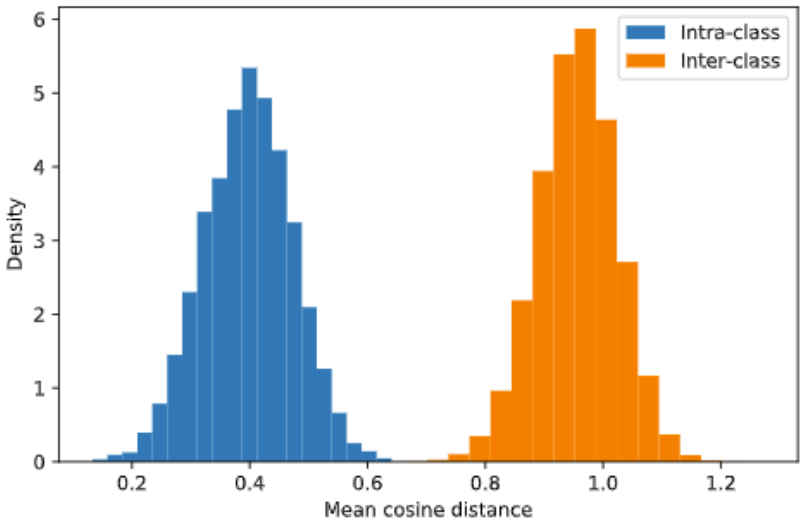
- Models trained only on `highres` (32 and 144) perform very poorly due to the small amount of data.
- Best model `highlow144` must be analyzed cautiously: Gain in accuracy is less than 1% which can be purely random as well.
- Model `lowres32` is still competitive $\rightarrow$ generalizes well.
- Baseline models `lowres32` and `lowres144` are the 2nd and 3rd best performing models, without any sort of fine-tuning.
- Entirely freezing backbones (training the head only) $\rightarrow$ poorer performances with fine-tuning than unfreezing the Output layers.

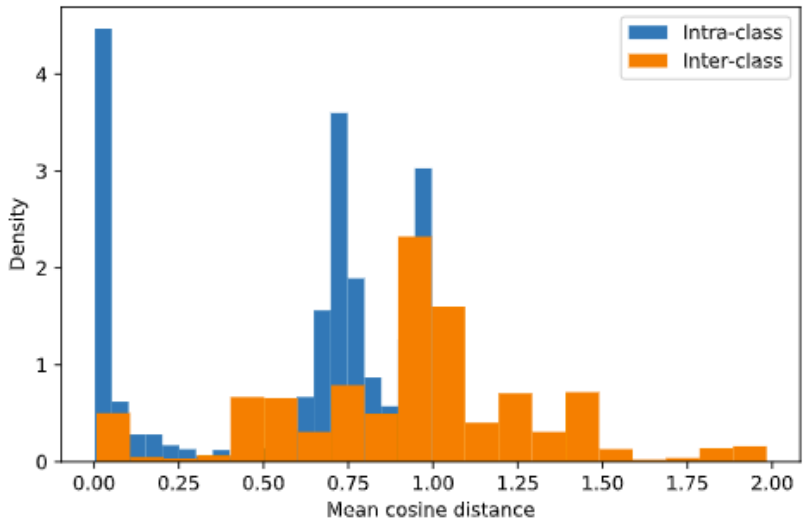
Embedding space analysis

- Is there an adaptation problem where the network optimizes the two different resolutions images separately?
- Are embeddings properly placed in the desired space? For instance, in case we reduce the embedding dimension to 3, are the embeddings placed in a sphere and are they occupying the whole sphere? (Should be when using ArcFace).
- Are same identities embeddings close? Are different identities embeddings far away?
- In case a network has been trained solely on low resolution images up-scaled, is an up-scaled low resolution image embedding in the same space than an high resolution image embedding?
- If we fine-tune a network with another resolution dataset, are the embedding in the same space? If they are not in the same space, by how far? Can a linear transformation solve this problem?


X-axis
Y-axis
Z-axis


X-axis
Y-axis
Z-axis


Intra-class
Inter-class
Density
Mean cosine distance


Intra-class
Inter-class
Density
Mean cosine distance

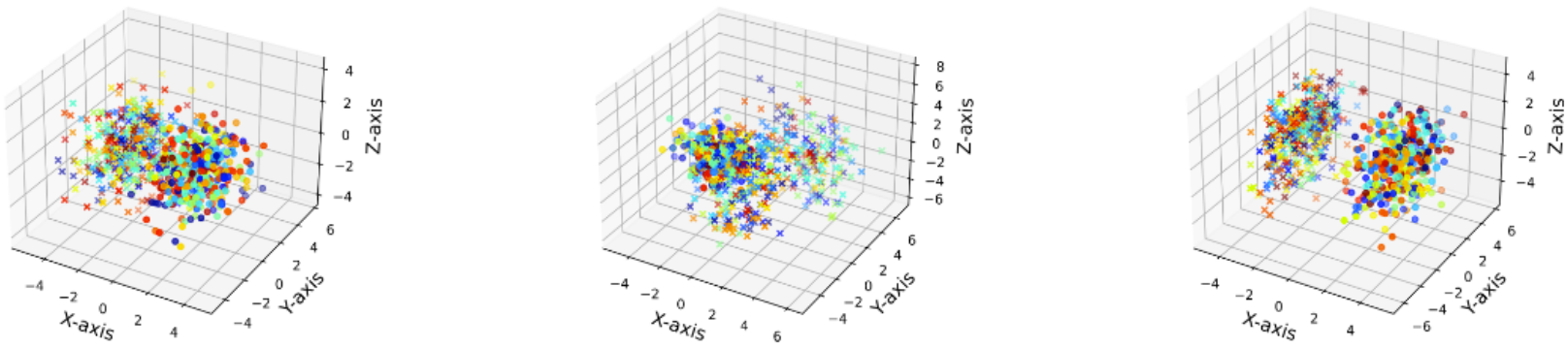


Figure 2: HR144 and HR144→32→144 in lowres144, lowres32 and highlow144

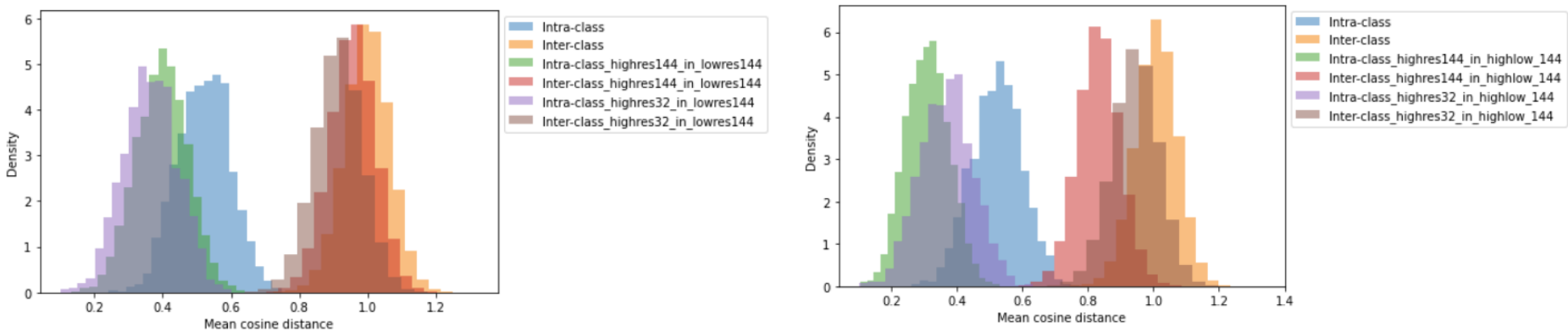


Figure 3: Histograms of pair-wise distance of HR144 and HR144→32→144 in lowres144 and highlow144

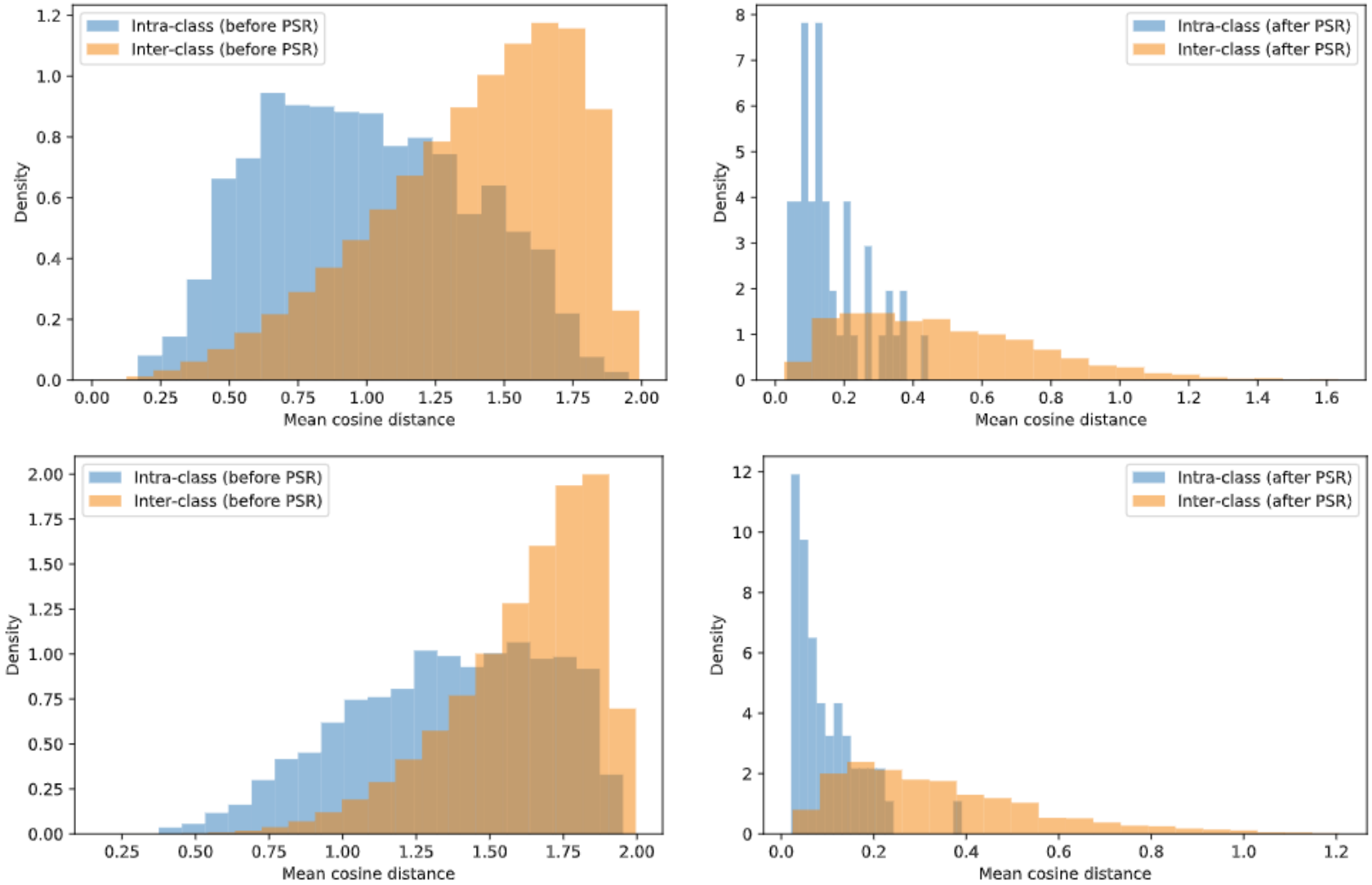


Figure 4: Histograms of cosine distance for lowres144 (top) and highlow (bottom) before and after Coherent Point Drift

Discussion

- Models can distinguish if the input is a blurred image and classify it accordingly, but also that in a mixed dataset with varying resolutions
- It is harder to distinguish each identity as the intra-class and inter-class distances overlap. This phenomenon is accentuated in Highlow144
- Coherent Point Drift is class agnostic: having a peak in intra-class distances after its application means that the spatial structure itself of the embeddings is similar

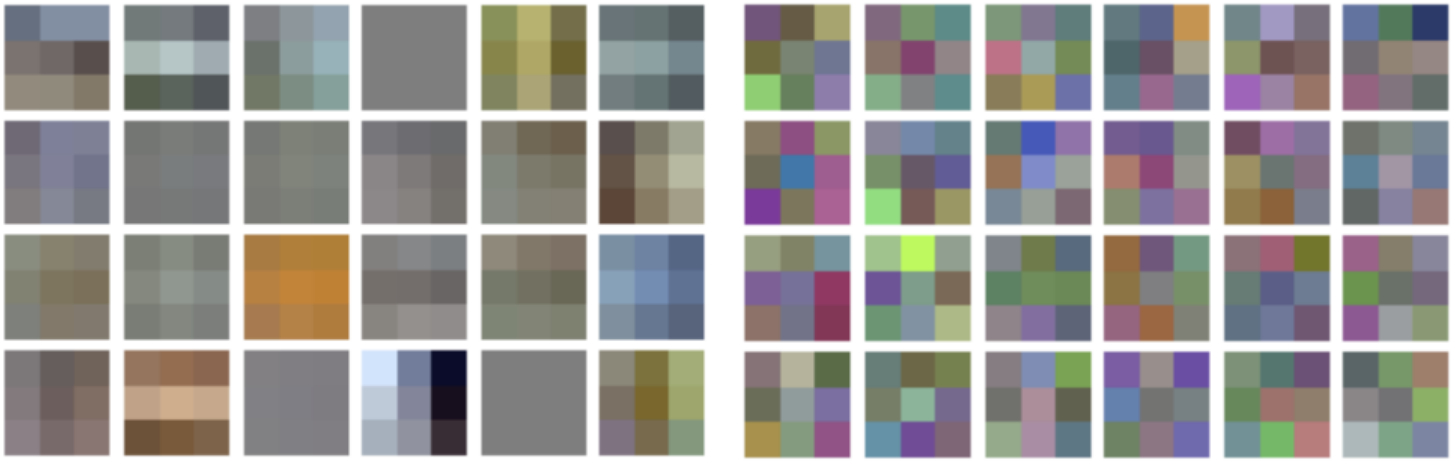


Figure 5: Few kernels for the first Conv2d of the Highlow144 (left) and Lowres144 (right)

Outline

1. Introduction
2. SOTA in HSFR
   - `cavaface.pytorch`
   - Angular loss
3. Proposed approach
   - Dataset
   - High volume-Low resolution training
   - Architecture
   - Training and Validation
4. Results
   - Baseline models
   - Fine-tuning
   - Embedding analysis
   - Weight analysis
5. Conclusion & extensions

Conclusion

- Addition of high resolution images does not seemed to improve the accuracy by a significant margin
- Separation of high and low resolution images into different regions. Through the use of transformation techniques, this separation can be drastically reduced which might be interesting in term of classification improvement
- Caveat: networks learns to distinguish images based on the "true" resolution which is a problem for high scale face recognition when multiple datasets of different sizes are to be mixed

Questions?

Acknowledgements

We want to particularly thank all IDEMIA team and especially our advisor Mr. Damien Monet which was a great help to us in so many ways. Even if the start of the project was difficult due to limited computational resources, we were able to properly run the models and gather valuable insights and recommendations thanks to IDEMIA's help.

This project subjet was really interesting and allowed us to see classification in a different manner through the usage of other and more sophisticated architectures and losses.

Finally, we would like to thank our master directors, Mrs. Marie-Paul Cani and Mr. Erwan Scornet that made this project possible.

References I

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition, 2019. URL https://ieeexplore.ieee.org/document/8953658.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. URL https://ieeexplore.ieee.org/document/7780459.

[3] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu. Squeeze-and-excitation networks, 2019. URL https://ieeexplore.ieee.org/document/8701503.

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

[5] S. Khallaghi. Python-CPD, 2020. URL https://github.com/siavashk/pycpd.

References II

[6] Y. Li and L. Chi. cavaface.pytorch: A Pytorch Training Framework for Deep Face Recognition, 2020. URL https://github.com/cavalleria/cavaface.pytorch.

[7] A. Myronenko and X. B. Song. Point-set registration: Coherent point drift. *CoRR*, abs/0905.2635, 2009. URL http://arxiv.org/abs/0905.2635.

[8] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. CosFace: Large Margin Cosine Loss for Deep Face Recognition, 2018. URL https://ieeexplore.ieee.org/document/8578650.

[9] C. Wei. Insight Face in TensorFlow, 2018. URL https://github.com/auroua/InsightFace_TF.