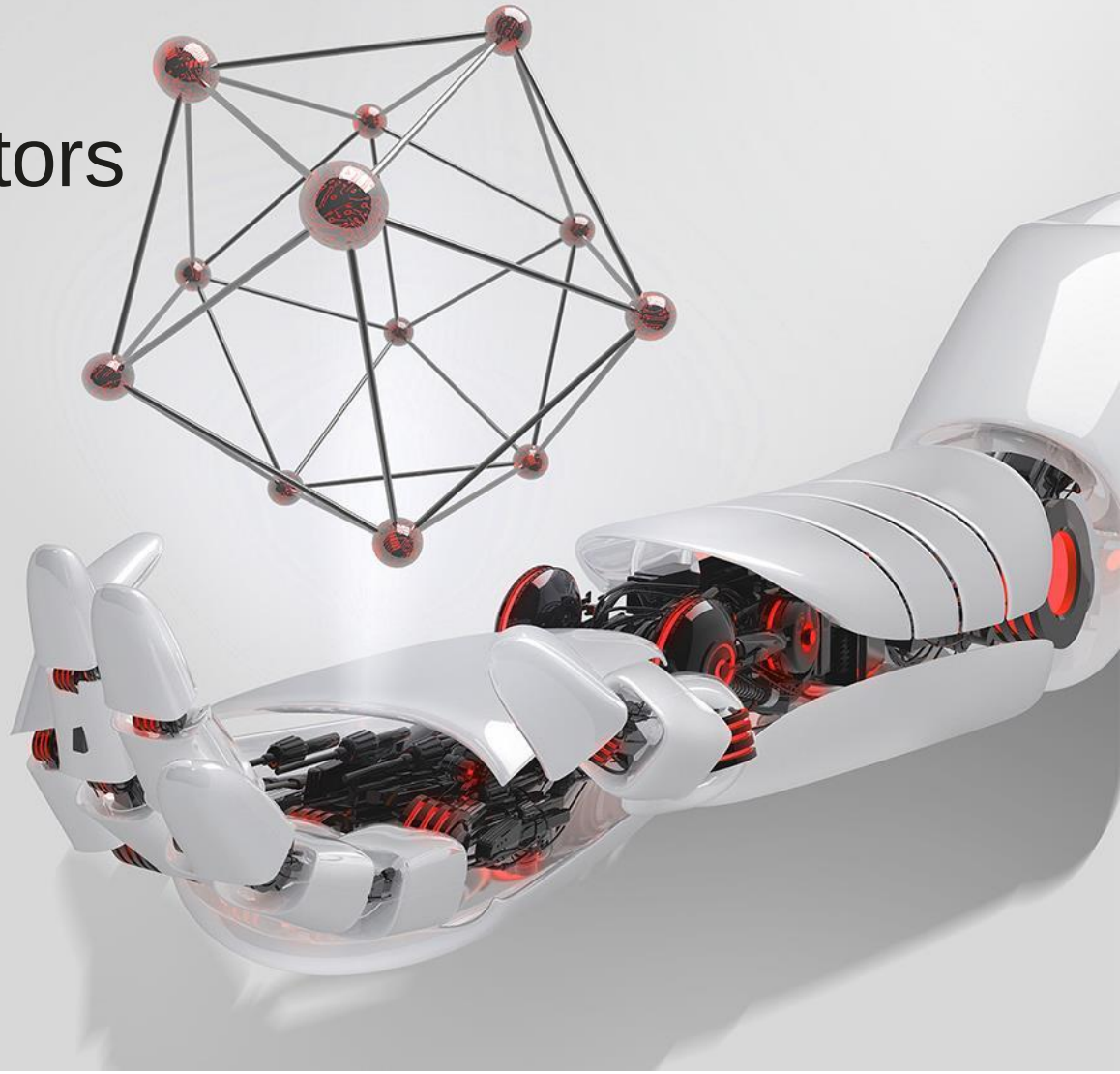
Metric learning for facial descriptors

Department: name: **Intelligent System Lab**
Author's name: **Petyushko Alexander, Komkov Stepan**
Date: **4th of July, 2019**



Security Level:

RAAI Summer School 2019

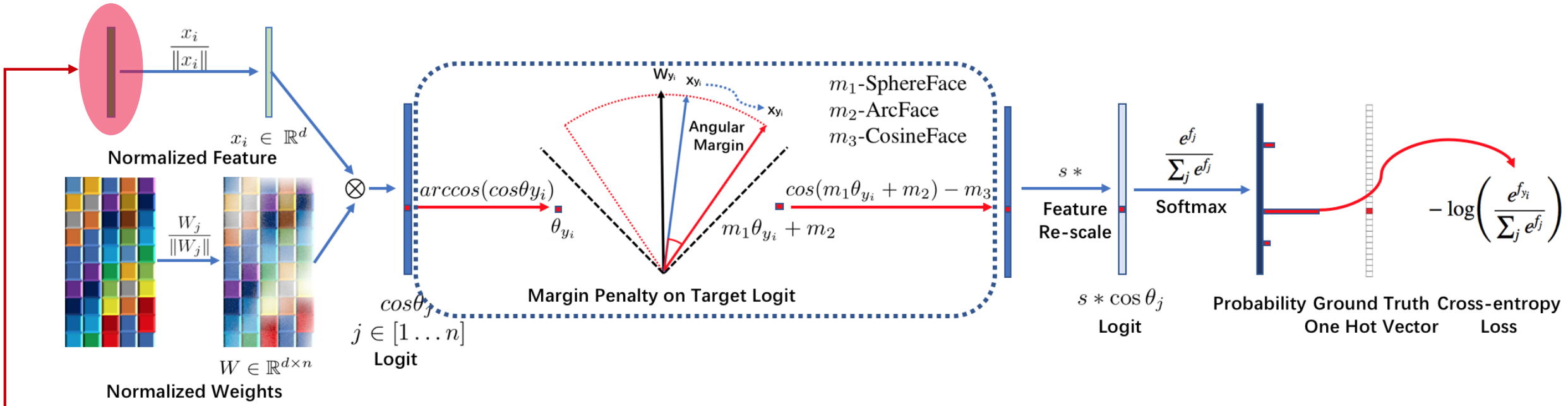
Developing artificial don't overlook natural

HUAWEI

ArcFace - Face ID one of the best public solution

Jan 23, 2018: a paper with novel ArcLoss loss function was published

“ArcFace: Additive Angular Margin Loss for Deep Face Recognition” by J. Deng, J. Guo, N. Xue and S. Zafeiriou



Don’ t worry: we are working with this embedding only

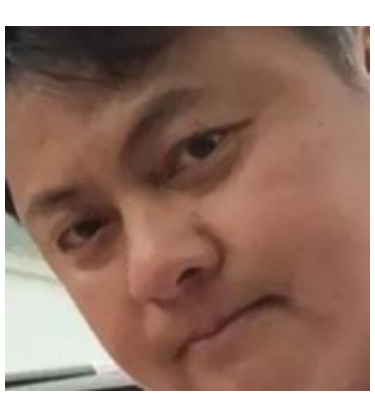
Case Study: Deep learning algorithms for Face Recognition

Standard pipeline:

1) Face detection (rectangular bounding box)
2) Landmarks detection (eyes, nose, mouth etc)
3) Face normalization (face warping to a predefined position)
4) Face feature (embedding) extraction using convolutional DNN
5) Usage of the similarity function depending on the specific task

Our area of interest as for now

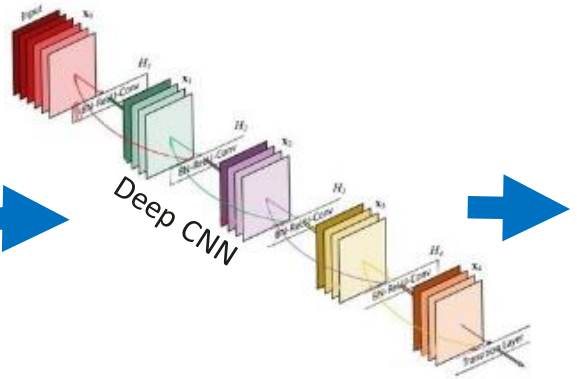
Also, some preprocessing / postprocessing etc…



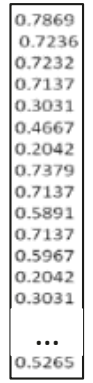
Input face

Face and Key points detection

Face alignment



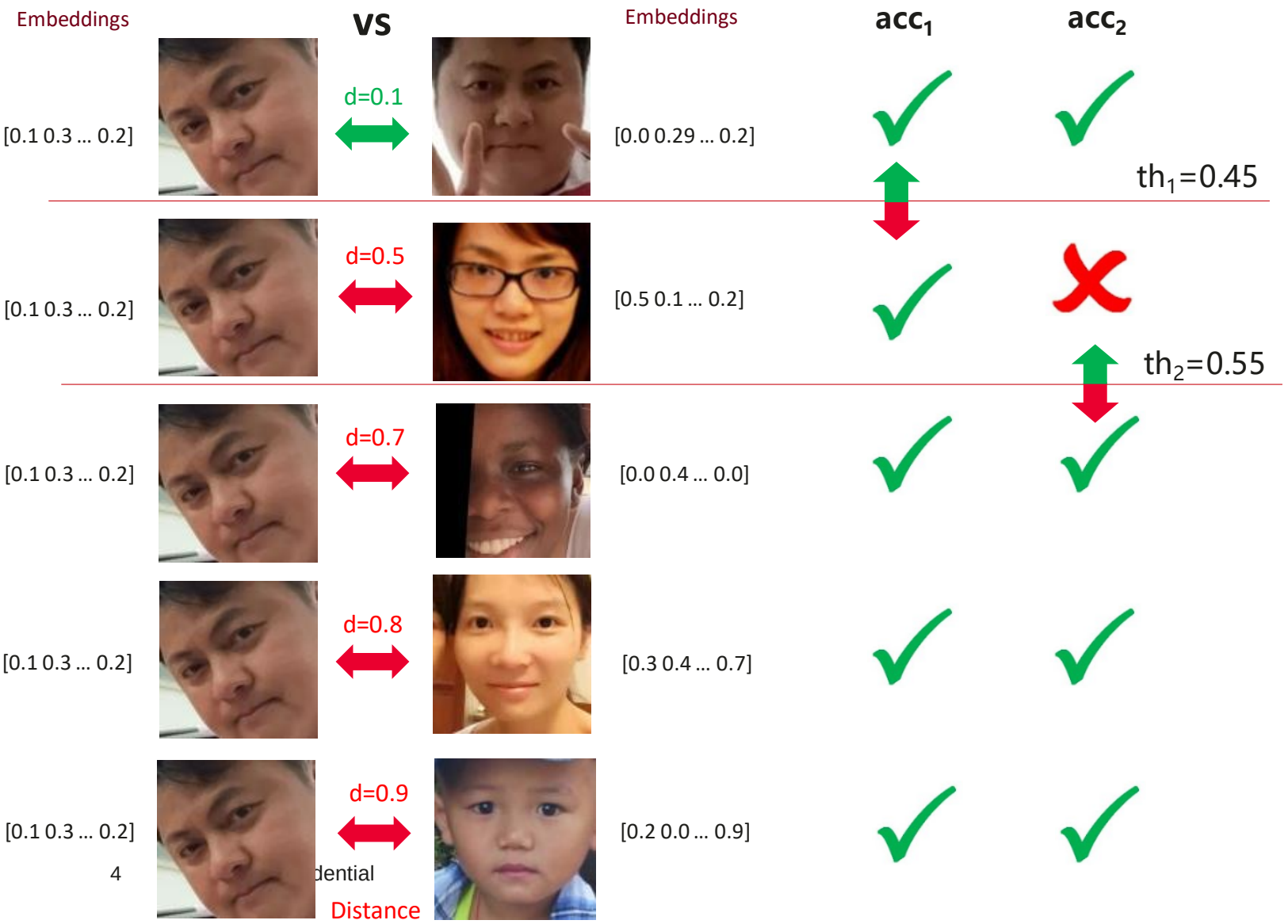
Feature extraction



Face features
[embedding]

HUAWEI

Case Study: Face Verification



Face Verification problem: design an algorithm to detect whether two input images belong to the same person

HUAWEI

Case Study: Face Verification metric

Verification metric:

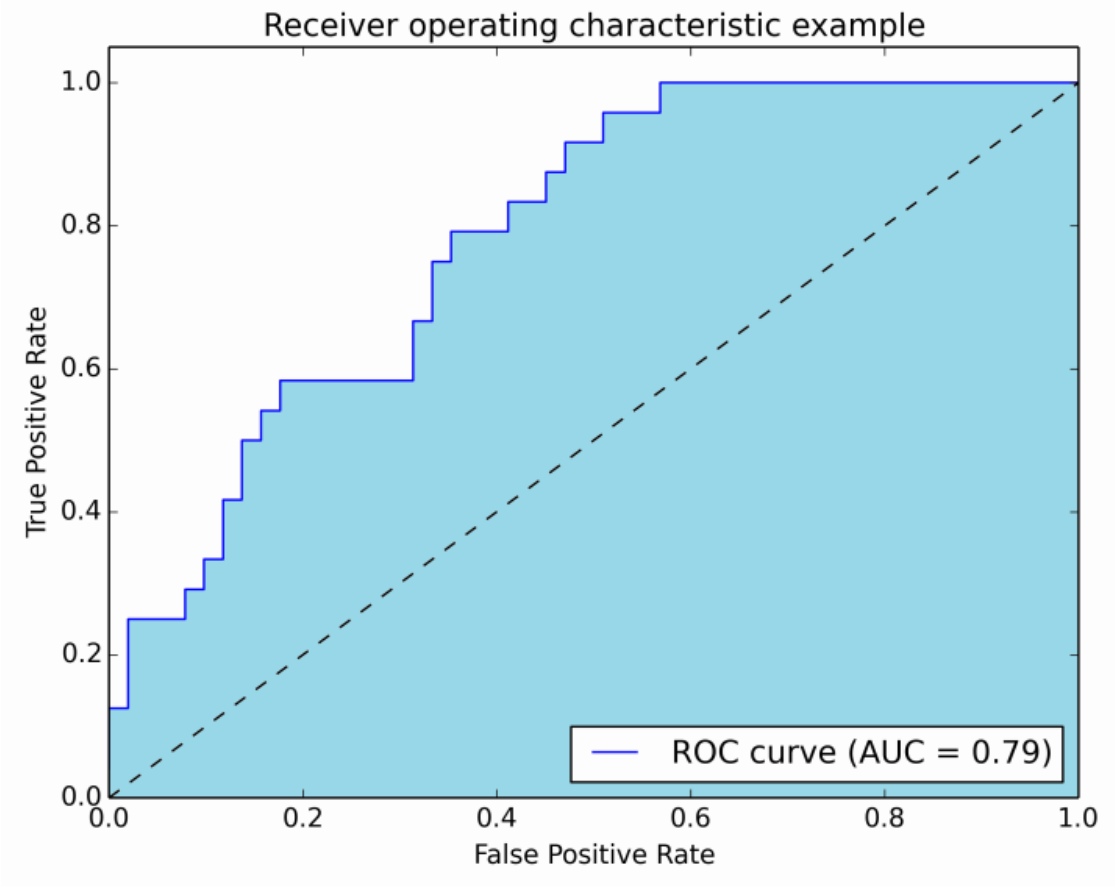
1) Receiver Operating Characteristic (ROC)
2) Area Under ROC (AUROC)

ROC:

1) X axis: False Positive Rate (FPR) – ratio of “different” pairs recognized as the same person
2) Y axis: True Positive Rate (TPR) – ratio of “same” pairs recognized as the same person
3) Each point corresponds to some threshold *th*

AUROC:

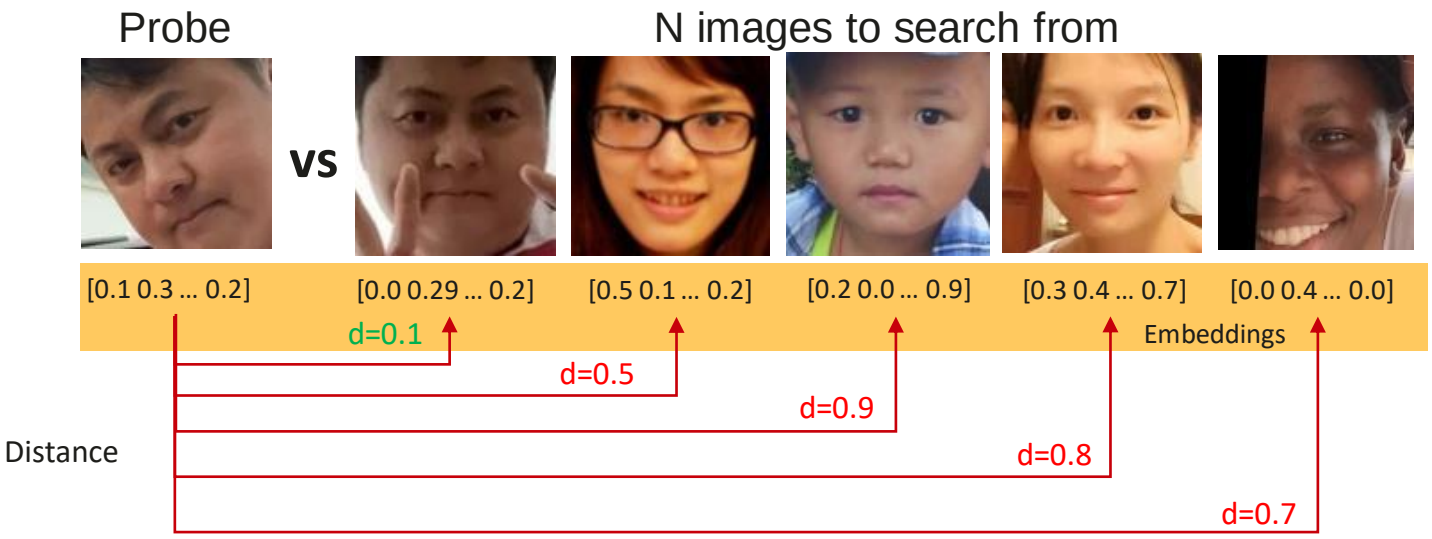
1) 0 <= AUROC <= 1
2) The more close to 1, the better

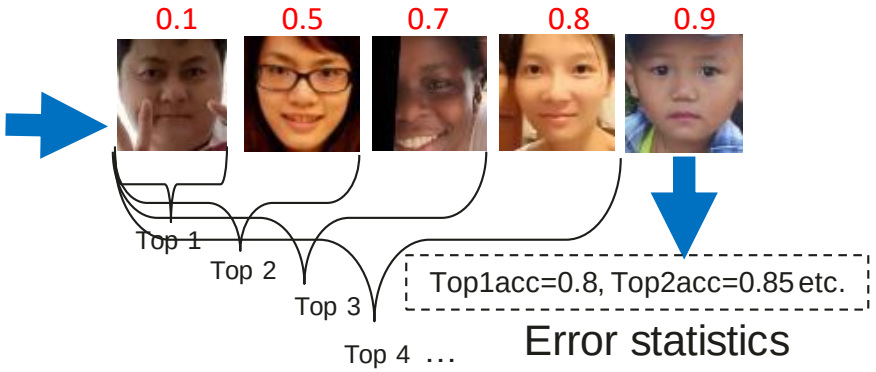


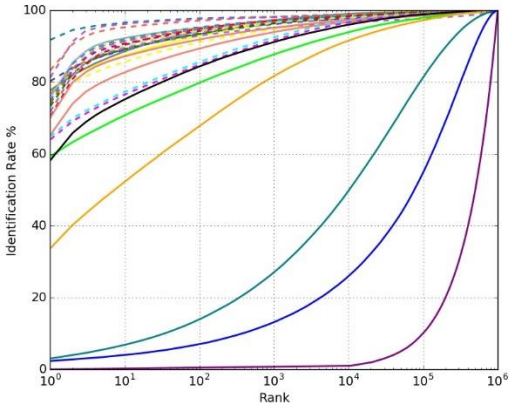
HUAWEI

Case Study: Face Identification

FaceID problem: design an algorithm to find a face of the given person presented by 1 image among N images





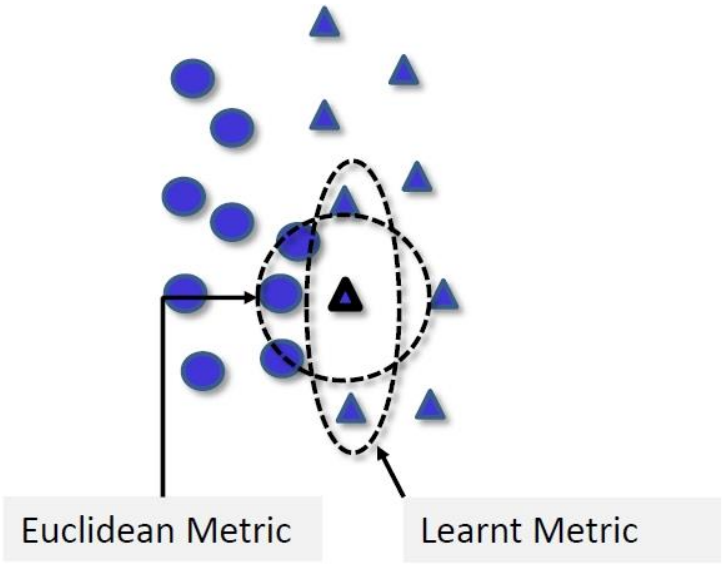


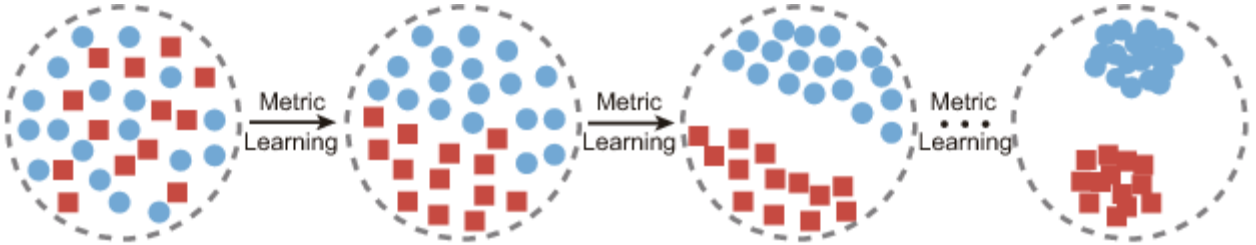
CMC metric is used
(not our current scope)

HUAWEI

Metric Learning

- Why **metric learning**?
  - Sometimes L_2 Euclidian distance is not enough
  - No need to retrain the model
  - No need to re-dump the embeddings
- What needs to be done?
  - Only slightly change the test procedure
  - Provide different distance / similarity function
    - Distance function is $d(x^1, x^2)$, where x^1, x^2 are input embeddings from R^N
    - The simplest case is $\boldsymbol{d(x^1, x^2) = \sum_{k=0}^{N-1}(x_k^1 - x_k^2)^2}$
- Main goal:
  - Make embeddings from the same person_id be closer than for different person_id





Task description

- **Embedding vector**: 512 elements

- **Train data (if needed)**:
  - train_labels.csv
    - Columns: index, class
    - index: unique counter (0..454713)
    - Class: person_id (0..10574)
  - train_set.npy
    - Matrix: 454714 x 512

Task description

- **Test data**:
  - test_pairs.csv
    - Columns: index1,index2,Id
    - index1,2: rows number from test embedding DB (0..11604)
    - Id: unique counter (0..31443)
  - test_set.npy
    - Matrix: 11605 x 512

Task statement

- **Task**
  - 1) Implement another metric function different from Euclidian one: $\boldsymbol{d(x^1, x^2) \neq \sum_{k=0}^{N-1}(x_k^1 - x_k^2)^2}$
  - 2) Dump to csv the distance for every pair listed in test_pairs.csv (you don't know whether this pair corresponds to the same person or not)
  - 3) Register on **kaggle** (if not done already) and submit your solution to it
  - 4) Register on **github** (if not done already) and make a pull request with the solutions better than the baseline (Euclidian distance)

Baseline code (Euclidian metric)

```
import numpy as np

import pandas as pd

pairs = pd.read_csv('test_pairs.csv')

test = np.load('test_set.npy')

distances = []

for i in range(len(pairs)):

    index1 = pairs['index1'][i]

    index2 = pairs['index2'][i]

    dist = np.sum((test[index1]-test[index2])**2)

    distances.append(dist)

baseline = pd.DataFrame({'Predicted':distances})

baseline['Id'] = range(len(baseline))

baseline.to_csv('baseline.csv',index=False)
```

Challenge notes

- **Important notes**
  - Finish time: **6th of July, 23:59**
  - Public portion: 25% of test
  - Private results will be available after "*Finish time*"
  - Public/Private portions have the same distribution
  - Number of submissions per day: 20 (max for kaggle)
  - Scored private submissions: 2

  - **1-3 places should prepare a couple of slides about their solution and present it on 7th of July**
  - Some prizes are expected!

Important links

- Kaggle
  - https://www.kaggle.com/c/metric-learning/overview

- Github (+ this presentation)
  - https://github.com/papermsucode/raai2019hackface

- Telegram chat
  - https://t.me/joinchat/B2iAGhBq7EaqwFtmPRelRw

Thank you.

Bring digital to every person, home, and organization for a fully connected, intelligent world.