國立高雄科技大學
National Kaohsiung University of Science and Technology

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff, Dmitry Kalenichenko, and James Philbin

IEEE Intl. Conf. on CVPR, 2015

Speaker: Shih-Shinh Huang

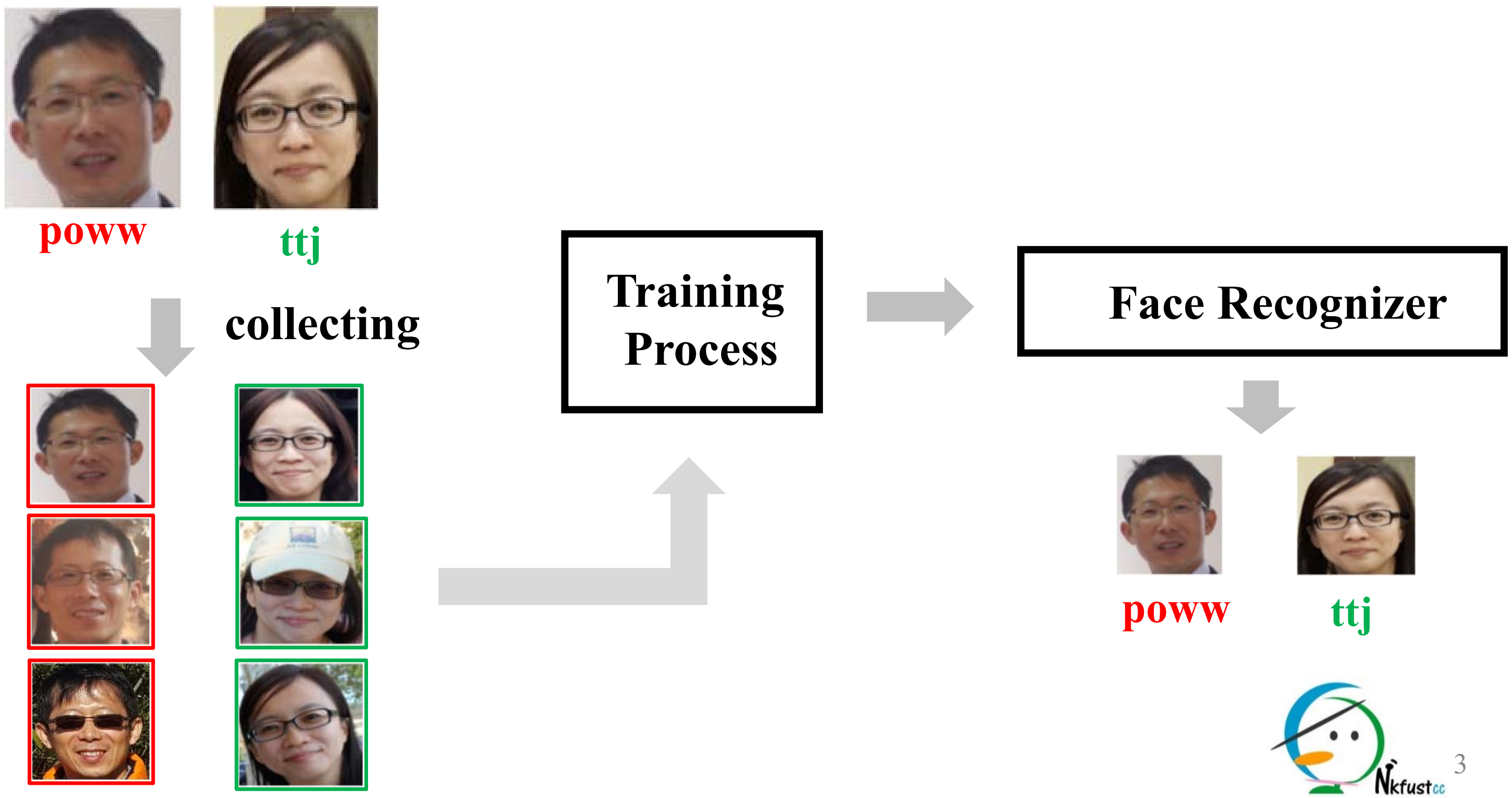
March 5, 2020

Outline

- Introduction
  - Motivation
  - Idea of FaceNet
  - Benefit from FaceNet
- Euclidean Embedding
  - FaceNet Architecture
  - Triplet Loss
- Triplet Selection
  - Motivation
  - Intuitive Selection
  - Online Generation

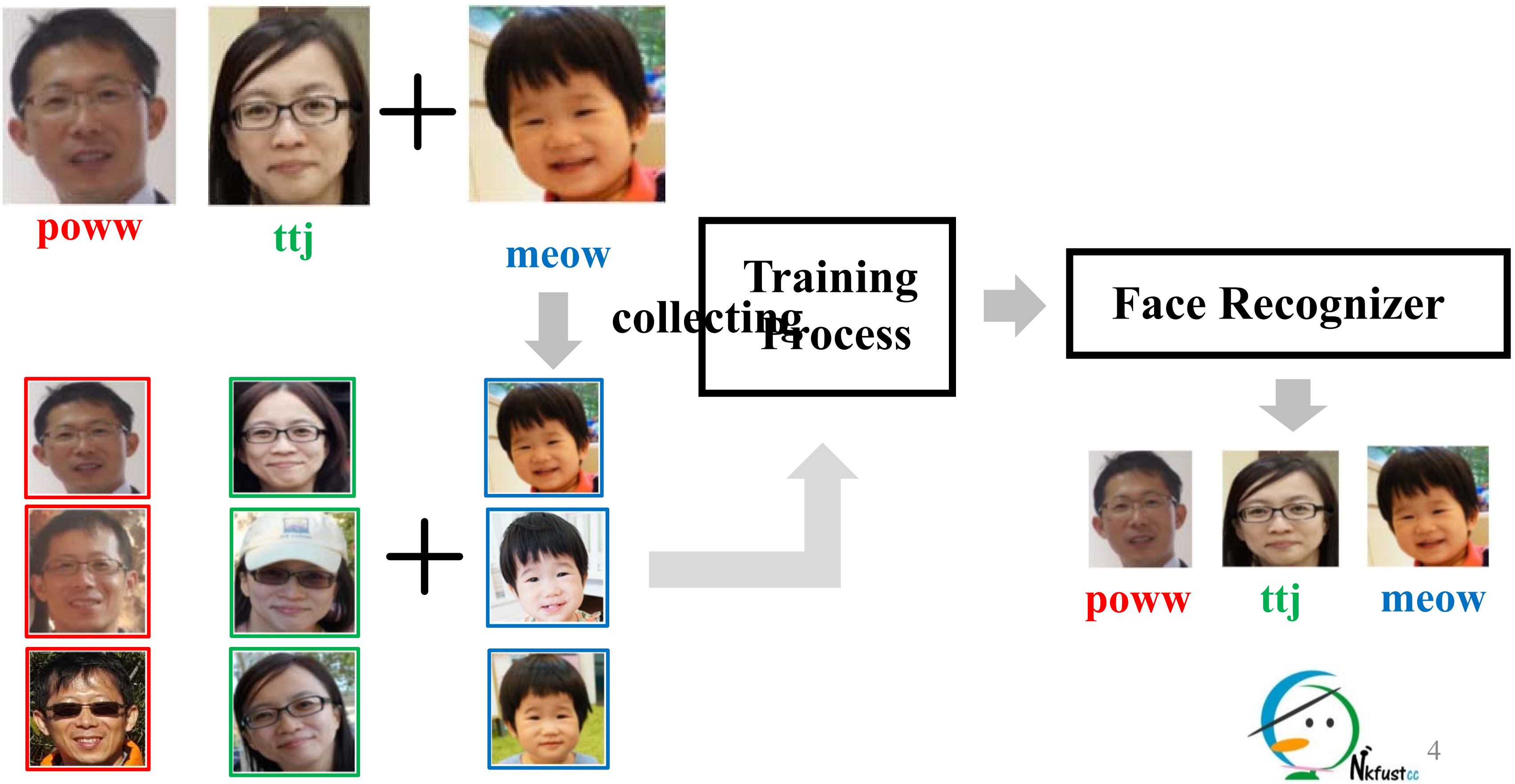
Introduction

- Motivation: Previous Face Recognition



Introduction

- Motivation: Previous Face Recognition



Introduction

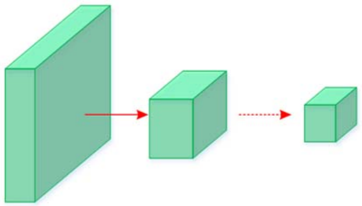
- Motivation: Problems
  - Retraining Problem: the cost of re-training is high and is sometimes not affordable.
  - Data Collection Problem: collecting sufficient faces to training a good classifier is impractical.

Introduction

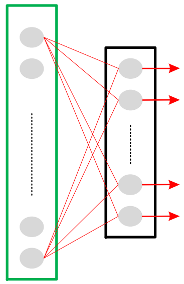
- Idea of FaceNet
  - A face recognizer generally consists of two stages
    - Feature Extraction: describe a face in an effective way.
    - Classification: assign an identifier to a face

Feature Extraction → Face Classification → poww

Deep Learning



convolution
neural network (CNN)



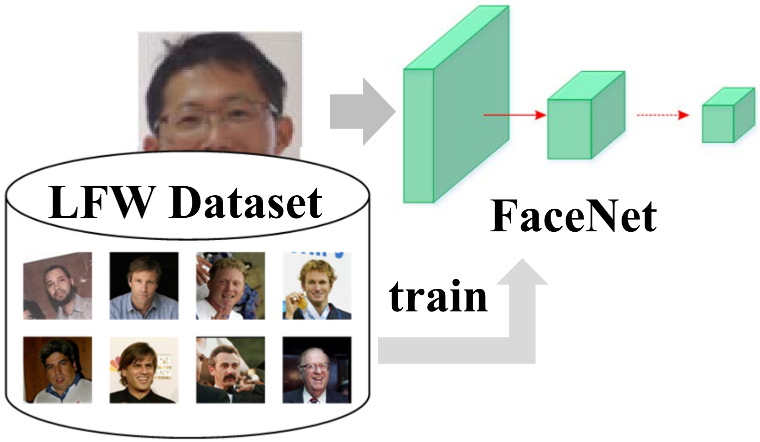
fully-connected
network

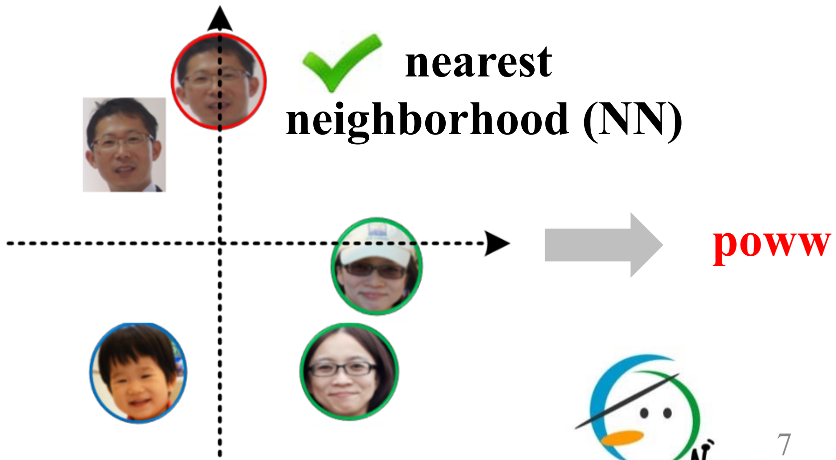
Introduction

- Idea of FaceNet
  - de-couple two face recognition stages

Feature Extraction → Face Classification

- FaceNet is a CNN just for feature extraction but not for classification.
- Face classification is achieved by using simple and training-free algorithms.

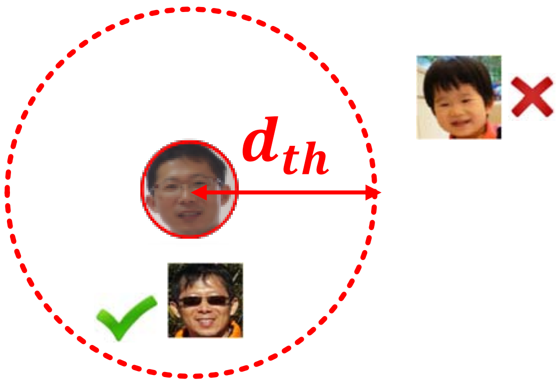




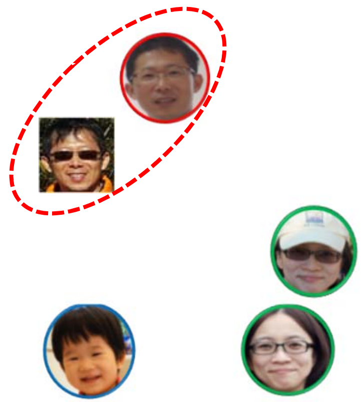
Introduction

- Benefit from FaceNet
  - The trained FaceNet is applicable to any face applications without re-training.
  - The use of FaceNet makes the number of faces necessary for face-related tasks small

verification → thresholding

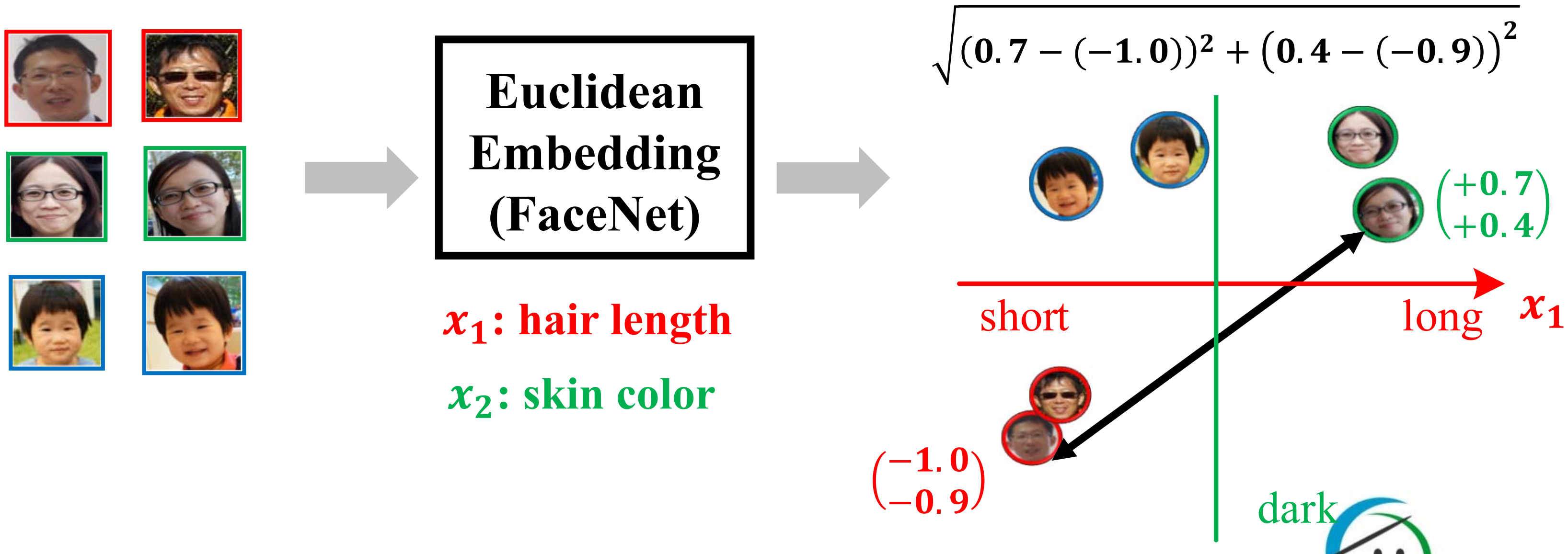


classification → nearest neighborhood



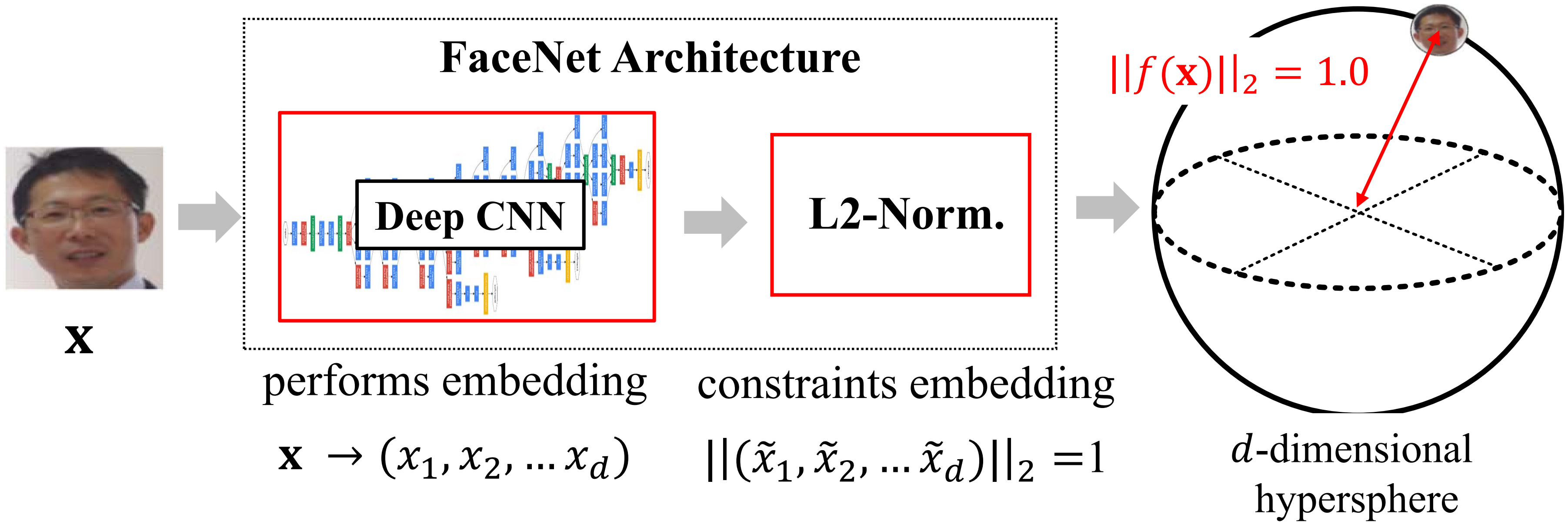
Euclidean Embedding

- FaceNet Architecture $f(.)$
  - maps a face clip to Euclidean space
  - makes the similarity measure possible.



Euclidean Embedding

- FaceNet Architecture $f(.)$
  - Formal Description: maps a face $\mathbf{x}$ to a d-dimensional unit vector $f(\mathbf{x}) = (\tilde{x}_1, \tilde{x}_2, \dots \tilde{x}_d)$



Euclidean Embedding

- FaceNet Architecture $f(.)$
  - Formal Description: maps a face $\mathbf{x}$ to a d-dimensional unit vector $f(\mathbf{x}) = (\tilde{x}_1, \tilde{x}_2, \dots \tilde{x}_d)$

FaceNet Architecture

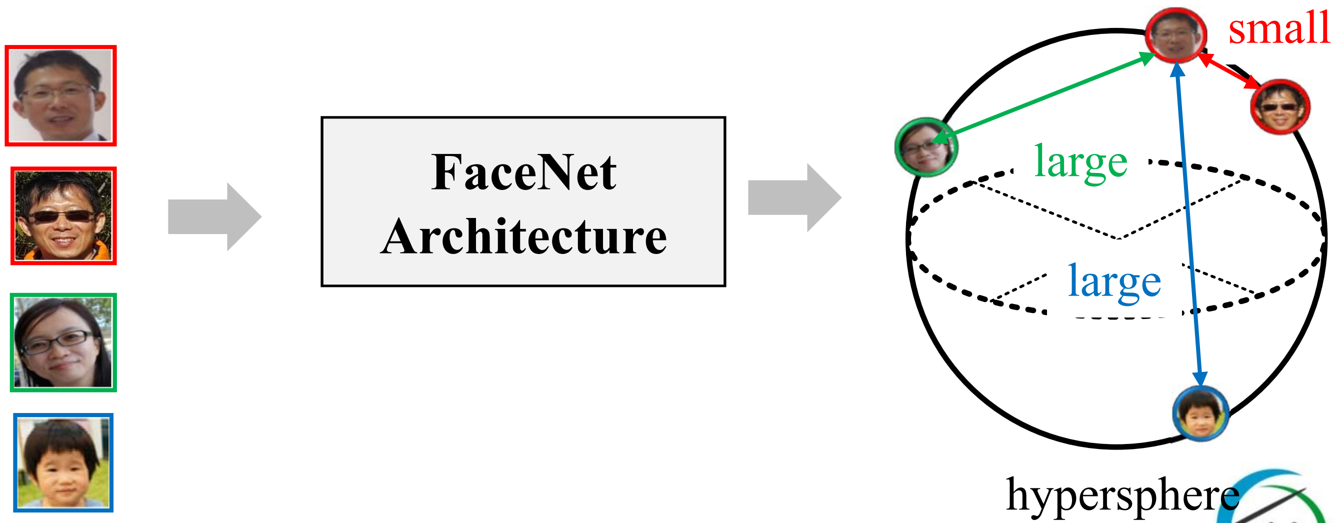
Deep CNN (d=128)
- Style Network
- Inception Type Network

L2-Norm.

- M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," CoPR, 2013
- C. Szegedy, et. al. "Going Deeper with Convolutions," CoPR, 2014

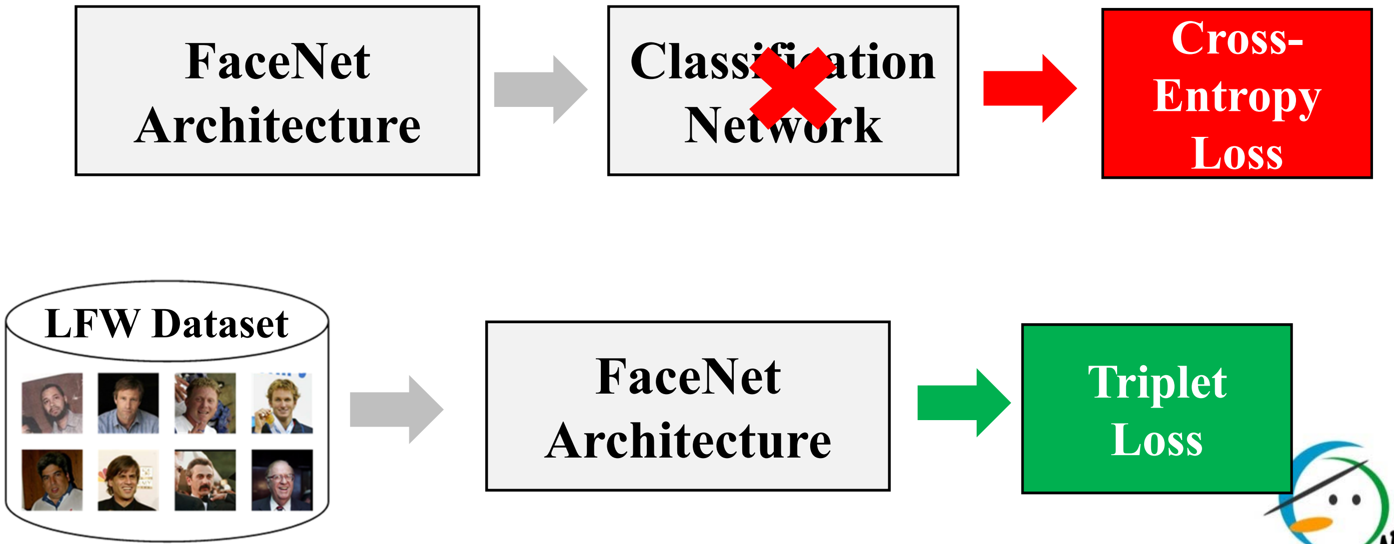
Euclidean Embedding

- FaceNet Architecture $f(.)$
  - Objective: provide discriminative embedding
    - faces of the same person have small distances
    - faces of distinct persons have large distances



Euclidean Embedding

- Triplet Loss
  - It is a loss defined in embedding feature space.
    - be independent of faces to be recognized
    - be trained by using off-the-shelf face datasets



Euclidean Embedding

- Triplet Loss
  - models embedding discriminability on a triplet $(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$

$\mathbf{x}^a$
anchor image

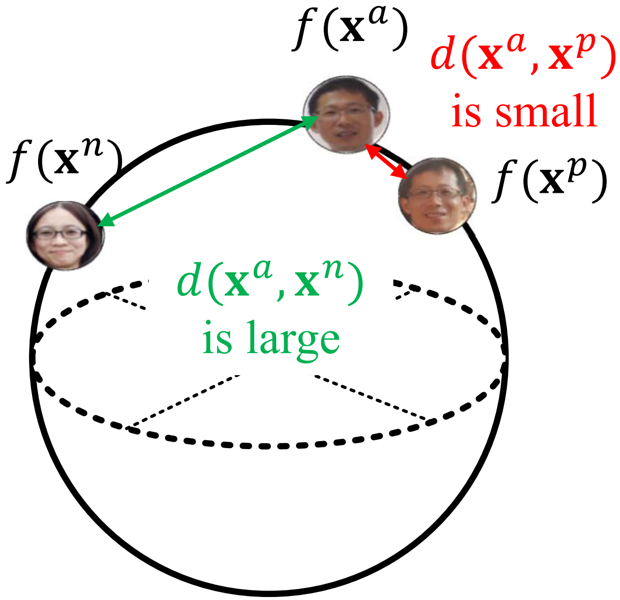
$\mathbf{x}^p$
positive image

$\mathbf{x}^n$
negative image

FaceNet Architecture

$$d(\mathbf{x}^a, \mathbf{x}^p) = \| f(\mathbf{x}^a) - f(\mathbf{x}^p) \|_2^2$$

$$d(\mathbf{x}^a, \mathbf{x}^n) = \| f(\mathbf{x}^a) - f(\mathbf{x}^n) \|_2^2$$

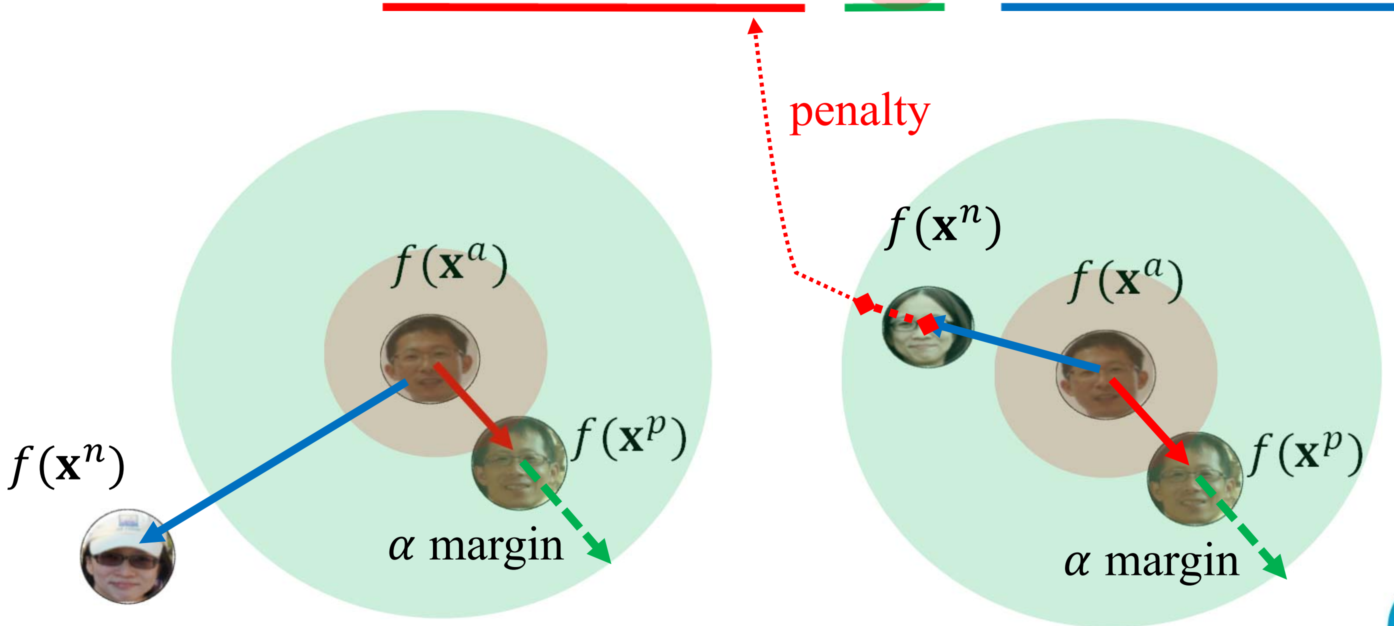


Euclidean Embedding

- Triplet Loss

$$Loss(\mathbf{x}^a, \mathbf{x}^p, \mathbf{x}^n)$$

$$= \max\{d(f(\mathbf{x}^a), f(\mathbf{x}^p)) + \alpha - d(f(\mathbf{x}^a), f(\mathbf{x}^n)), \quad 0\}$$

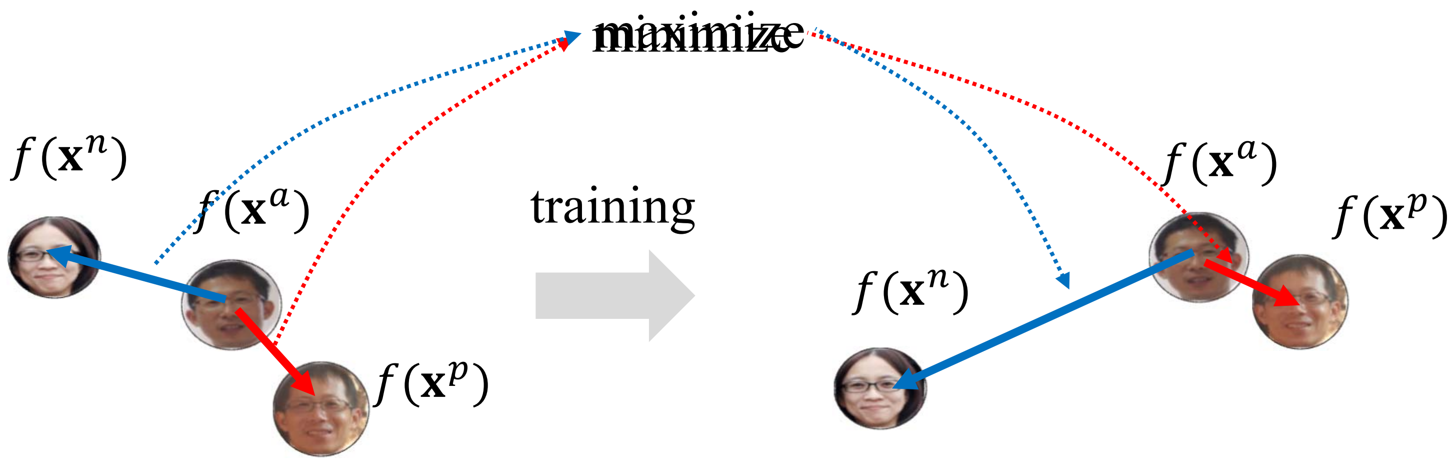


Euclidean Embedding

- Triplet Loss

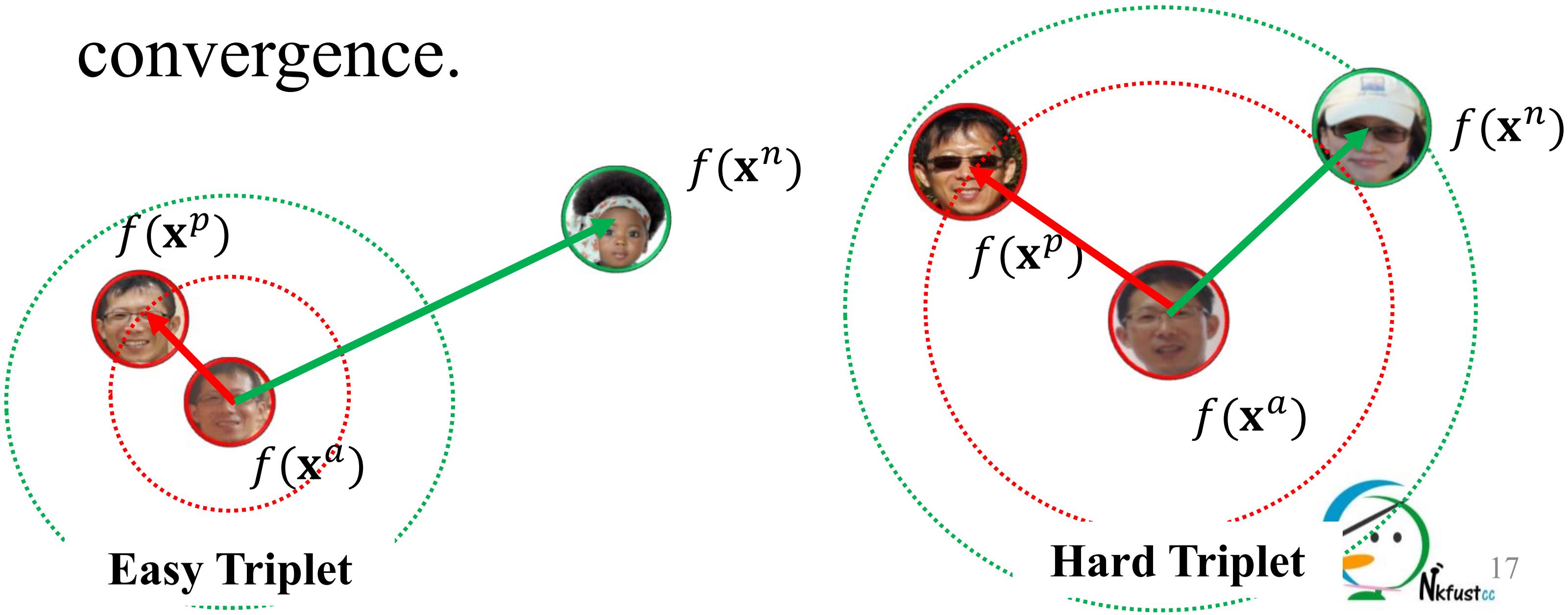
$$L(T) = \sum_{i \in T} Loss(\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{x}_i^n)$$

set of all triplets



Triplet Selection

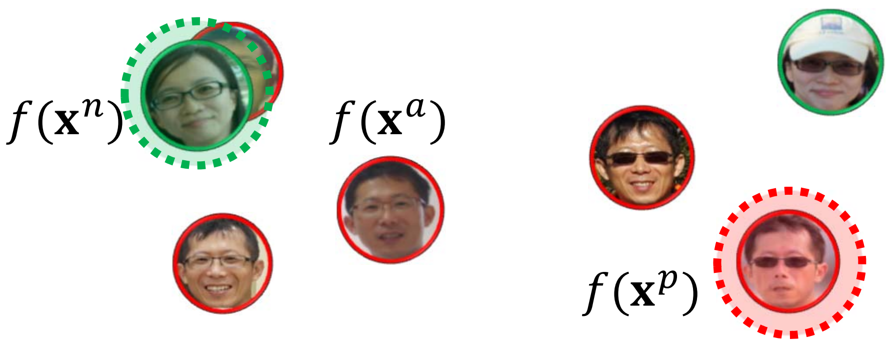
- Motivation
  - Using all possible triples for training leads to slow convergence.
  - It selects hard triplets for training to ensure fast convergence.



Triplet Selection

- Intuitive Selection
  - find the hardest positive $\hat{\mathbf{x}}^p$ and negative $\hat{\mathbf{x}}^n$ to an anchor $\mathbf{x}^{\mathrm{a}}$.

$$\hat{\mathbf{x}}^p = \arg\max_{\mathbf{x}^p} ||f(\mathbf{x}^{\mathrm{a}}) - f(\mathbf{x}^p)|| \qquad \hat{\mathbf{x}}^n = \arg\min_{\mathbf{x}^n} ||f(\mathbf{x}^{\mathrm{a}}) - f(\mathbf{x}^n)||$$

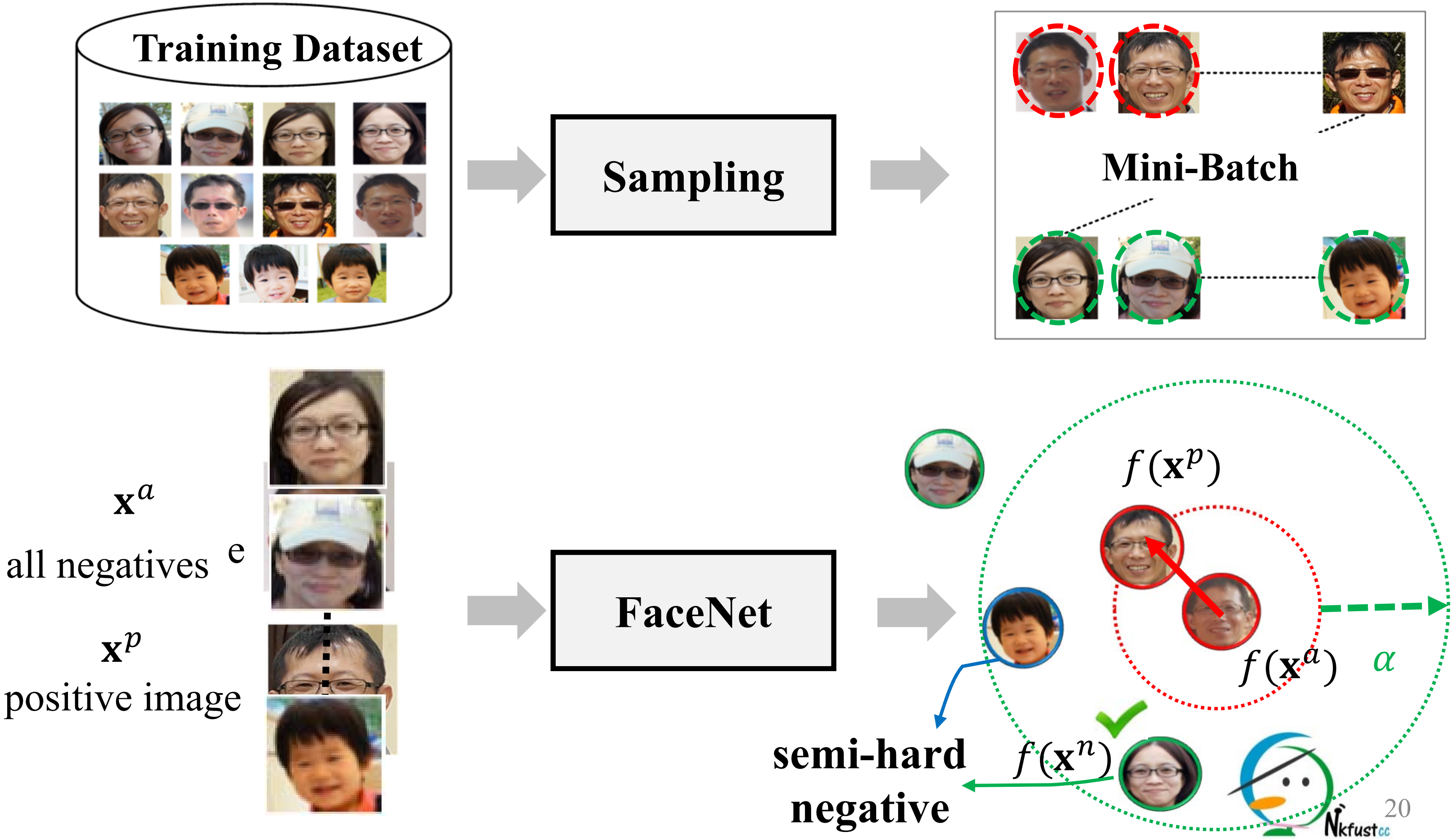


drawback: poor training
reason: poorly imaged faces would dominate the selection

Triplet Selection

- Online Generation
  - mini-batch generation: sample training dataset to generate mini-batch around 1800 faces
    - around 40 faces per identity
    - randomly sampled negative faces
  - triplet formation: form all possible anchor-positive-negative triplets within the mini-batch.
    - use all anchor-positive pairs in mini-batch
    - randomly take a semi-hard negative (within margin)

Triplet Selection




thank you
© 1996, 2002 SANRIO CO., LTD.