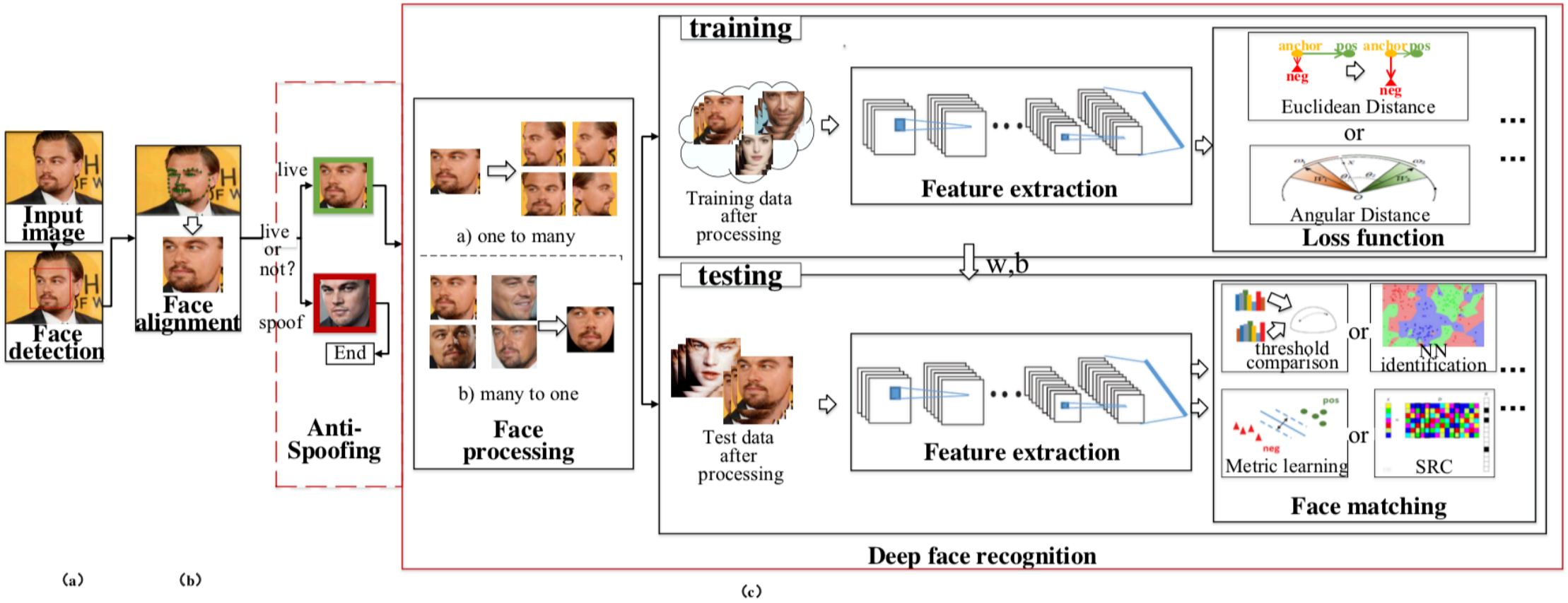
FaceNet:

A Unified Embedding for Face Recognition and Clustering

M2019170
허성실

Background

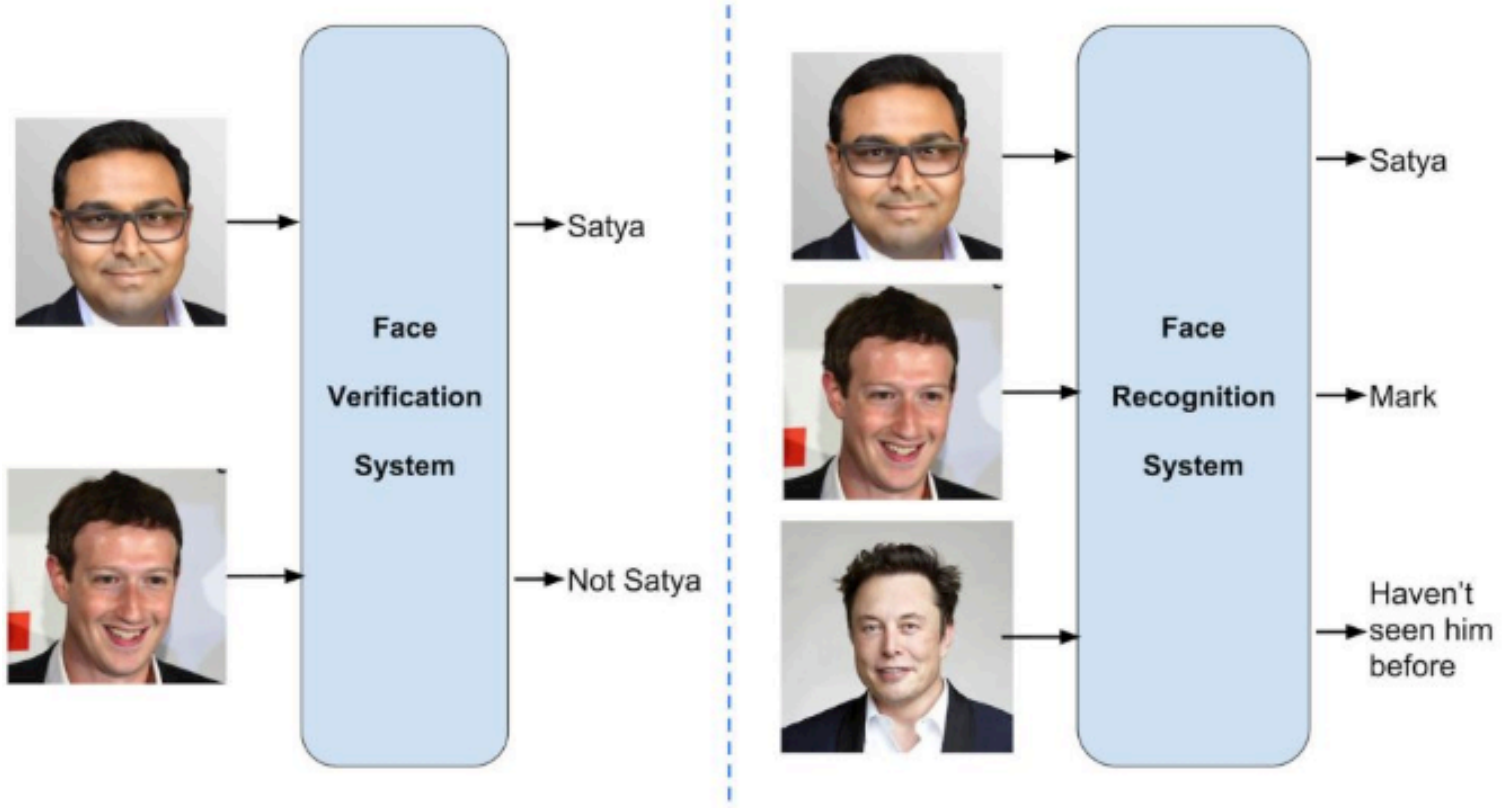
Face Recognition System



Face Verification & Identification

The Gallery: Target IDs
The Proble: Test ID

Verification: One-to-one
Identification: One-to-many

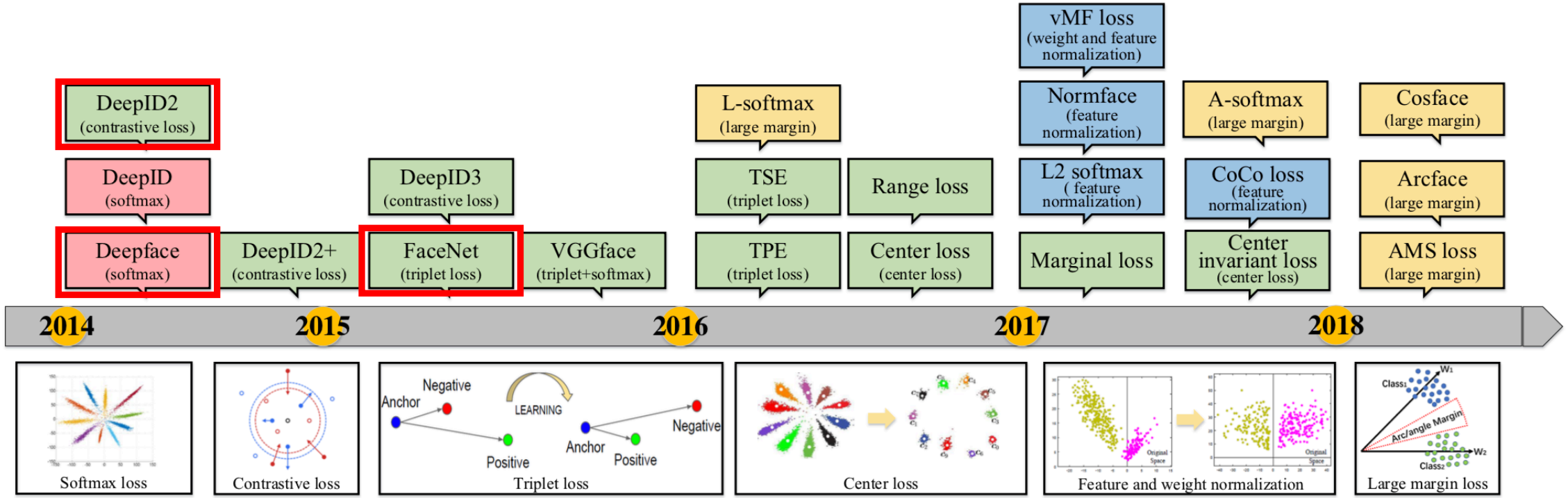


Face Verification

Face Identification(Recognition)

출처: https://www.learnopencv.com/face-recognition-an-introduction-for-beginners/

Face Recognition History

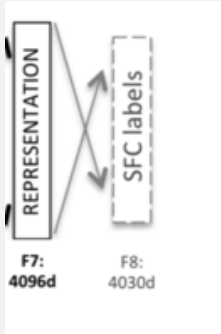


Deepface

i) Train

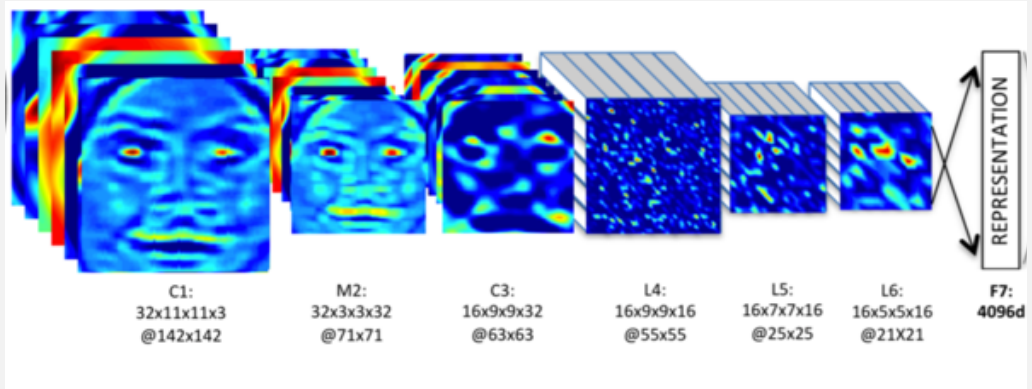
Detect & align

1. 얼굴 이미지들 간의 variation ↓
2. frontalized



classify

Softmax를 이용한 이미지 분류

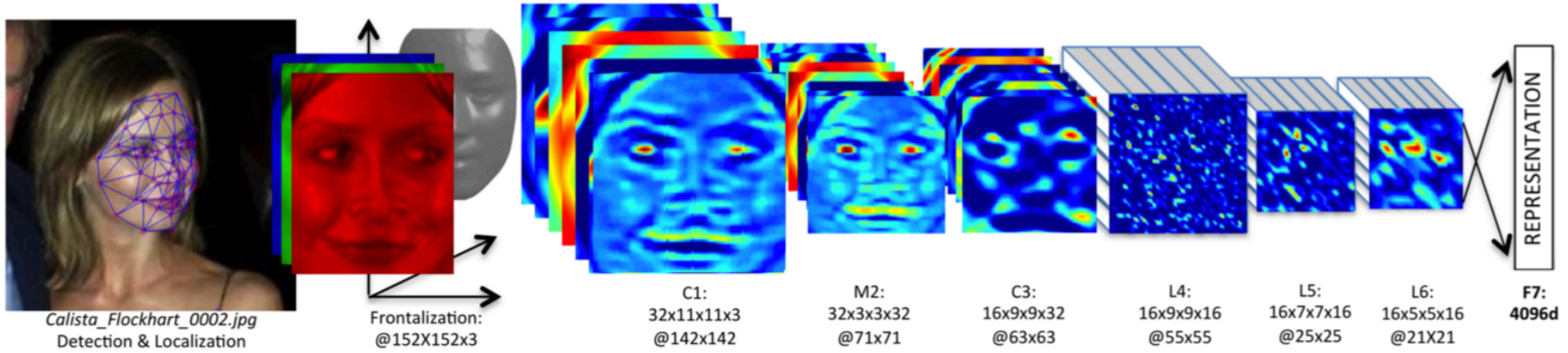


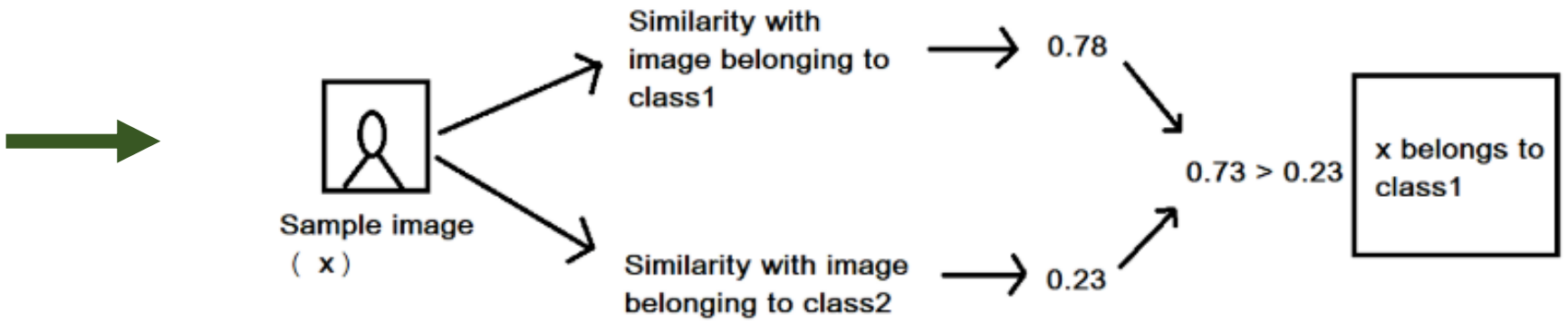
Represent

1. Low level feature 추출(c1, M2, C3)
2. Unshared weight을 이용한 학습(L4 ~L6)

Deepface

ii) Test





Siamese Network

출처: https://medium.com/predict/face-recognition-from-scratch-using-siamese-networks-and-tensorflow-df03e32f8cd0

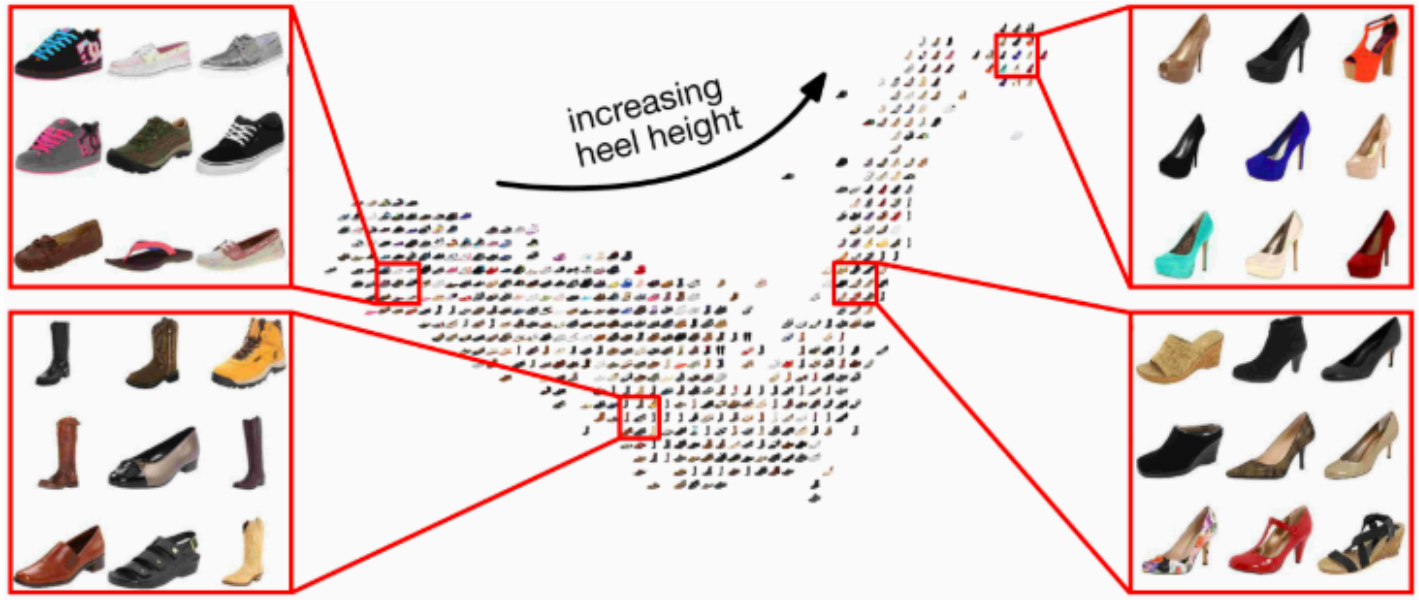
DeepID2

Deep Learning Face Representation by Joint Identification-Verification

Yi Sun[1] Xiaogang Wang[2] Xiaoou Tang[1,3]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]Department of Electronic Engineering, The Chinese University of Hong Kong
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
sy011@ie.cuhk.edu.hk xgwang@ee.cuhk.edu.hk xtang@ie.cuhk.edu.hk

- Metric Learning

  Metric learning learns a metric function from training data to calculate the similarity or distance between samples.



DeepID2

Loss function

Softmax + Verification

$$\text{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \begin{cases} \frac{1}{2}\left\|f_i - f_j\right\|_2^2 & \text{if } y_{ij} = 1 \\ \frac{1}{2}\max\left(0, m - \left\|f_i - f_j\right\|_2\right)^2 & \text{if } y_{ij} = -1 \end{cases}$$

DeepID2

- Fine-tune for Verification using Joint Bayesian (98.97 -> 99.15%)

결합 베이지안 모델(Joint Bayesian)

- 얼굴 특징 벡터: 얼굴의 내재적 특성 + 외재적 특성의 합으로 나타난다

$$f = \mu + \epsilon,$$

- μ와 ε은 정규분포를 따른다고 가정.
- 두 얼굴이 같을 확률 양수, 반대면 음수

$$\log \frac{P(f_1, f_2|H_{inter})}{P(f_1, f_2|H_{intra})}$$

Table 4: Accuracy comparison with the previous best results on LFW.

method	accuracy (%)
high-dim LBP [4]	95.17 ± 1.13
TL Joint Bayesian [2]	96.33 ± 1.08
DeepFace [22]	97.35 ± 0.25
DeepID [21]	97.45 ± 0.26
GaussianFace [14]	98.52 ± 0.66
DeepID2	99.15 ± 0.13

How about Facenet?

Method	Net. Loss	Outside data	# models	Aligned	Verif. metric	Layers	Accu.
DeepFace [97]	ident.	4M	4	3D	wt. chi-sq.	8	97.35±0.25
Canon. view CNN [115]	ident.	203K	60	2D	Jt. Bayes	7	96.45±0.25
DeepID [92]	ident.	203K	60	2D	Jt. Bayes	7	97.45±0.26
DeepID2 [88]	ident. + verif.	203K	25	2D	Jt. Bayes	7	99.15±0.13
DeepID2+ [93]	ident. + verif.	290K	25	2D	Jt. Bayes	7	99.47±0.12
DeepID3 [89]	ident. + verif.	290K	25	2D	Jt. Bayes	10-15	99.53±0.10
Face++ [113]	ident.	5M	1	2D	L2	10	99.50±0.36
FaceNet [82]	verif. (triplet)	260M	1	no	L2	22	99.60±0.09
Tencent [8]	-	1M	20	yes	Jt. Bayes	12	99.65±0.25

Table 2 CNN top results: As some of the highest results on LFW have been from using supervised convolutional neural networks (CNNs), we compare the details of the top-performing CNN methods in a separate table. N.B. – unknown parameters that were not mentioned in the corresponding papers are denoted with a "-".

- **Before FaceNet**
  - Training using Identification Loss (+ contrastive Loss)
  - Fine Tune using Metric Learning

vs

- **FaceNet**
  - Training using Metric Learning

Abstract

- **얼굴 이미지**에서 부터 **유클리디안 공간**으로 매핑하는 것을 학습

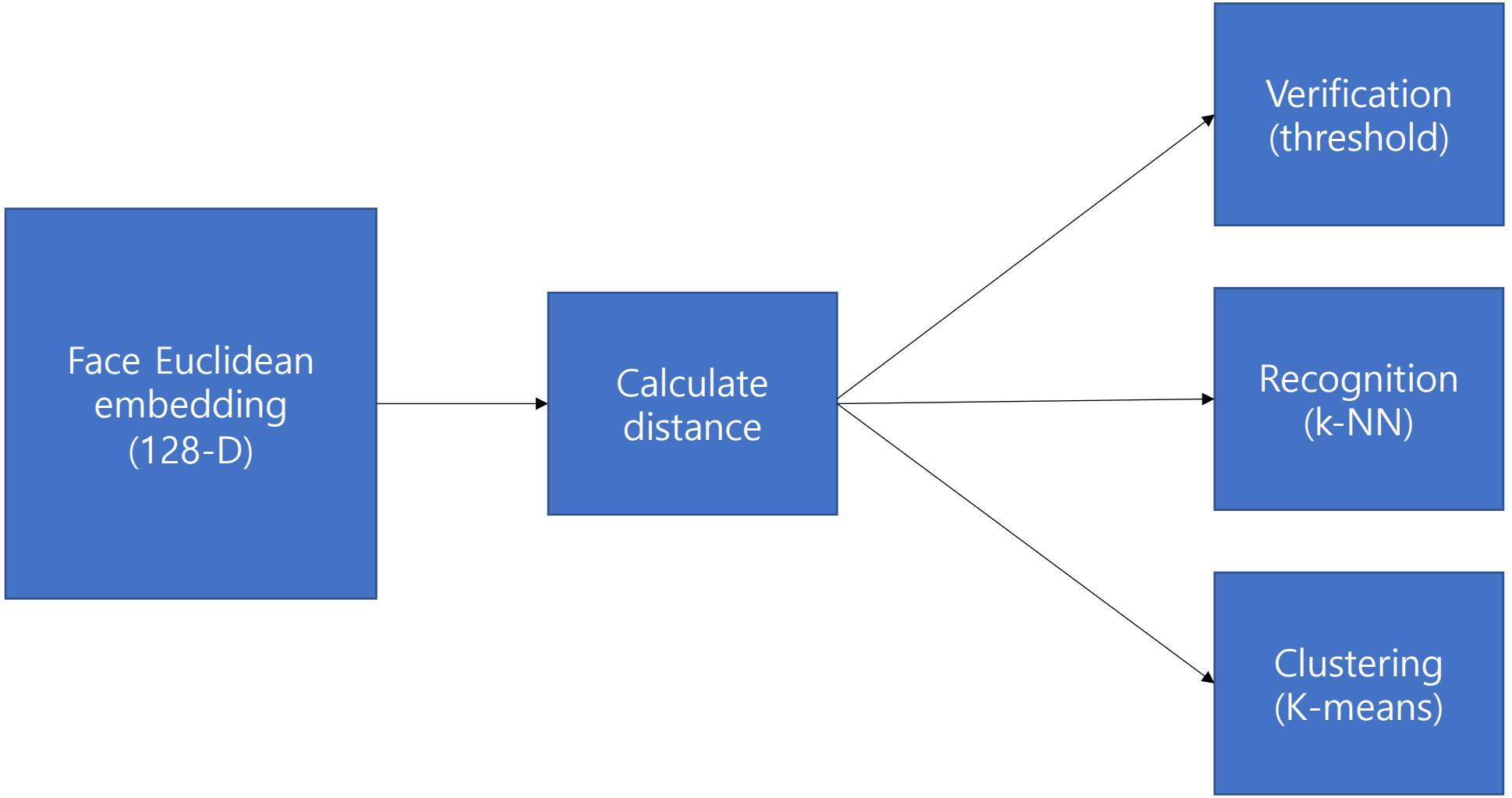
-> task(face recognition, verification and clustering) 수행이 쉬워짐

- 깊은 신경망을 사용하여 **직접** 임베딩하는 것을 최적화시킴

- aligned matching /non-matching 얼굴을 이용한 **triplets** 사용

- 정확도 : 99.63% (LFW dataset) & 95.12% (YouTube Faces DB)

Introduction


Face Euclidean embedding (128-D)
Calculate distance
Verification (threshold)
Recognition (k-NN)
Clustering (K-means)

- threshold = 1.1
- pose와 illumination이 있는 이미지에도 좋은 결과

- **이전 얼굴 인식용 네트워크**

- 중간에 bottleneck layers 를 사용
- 간접적 & 비효율적
- 그 외 (PCA로 벡터 축소, 2D or 3D alignment, Classification & Verification loss 혼합)

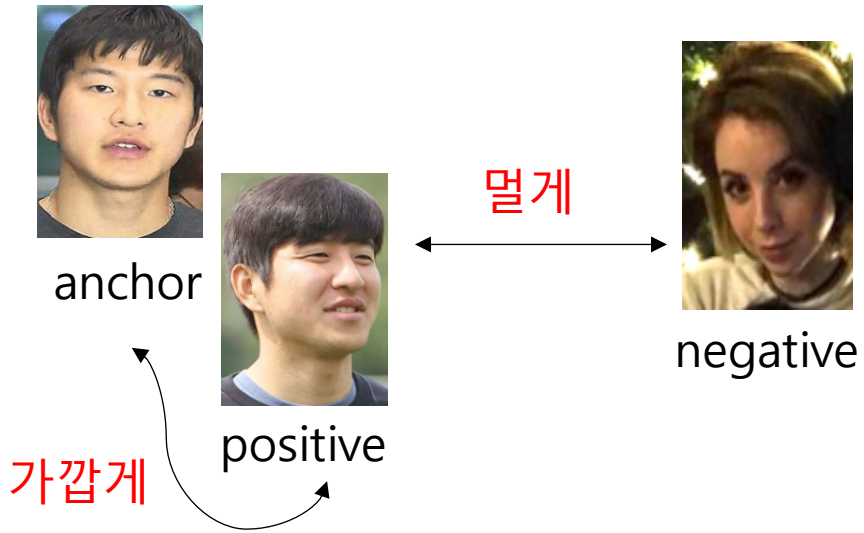
- **Facenet 얼굴 인식용 네트워크**

- Tripled-based loss function을 이용하여 직접 128차원 임베딩
- Loss는 같은 얼굴은 가깝게, 다른 얼굴은 멀게
- Novel online negative exemplar mining strategy 이용

Method

3.1 Triplet Loss

- 같으면 가깝게, 다르면 멀게



$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \ ,$$

$$\forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in \mathcal{T} \ .$$

3.1 Triplet Loss

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

- Hinge Loss: $[x]_+ = \max(0, x)$
  - Margin 최대화
- Constraint: $\|f(x)\|_2 = 1$
  - Euclidean space
  - d-dimensional hypersphere
- α=0.2로 고정

3.2 Triplet Loss

- triplet을 어떻게 고를까?

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \,,$$

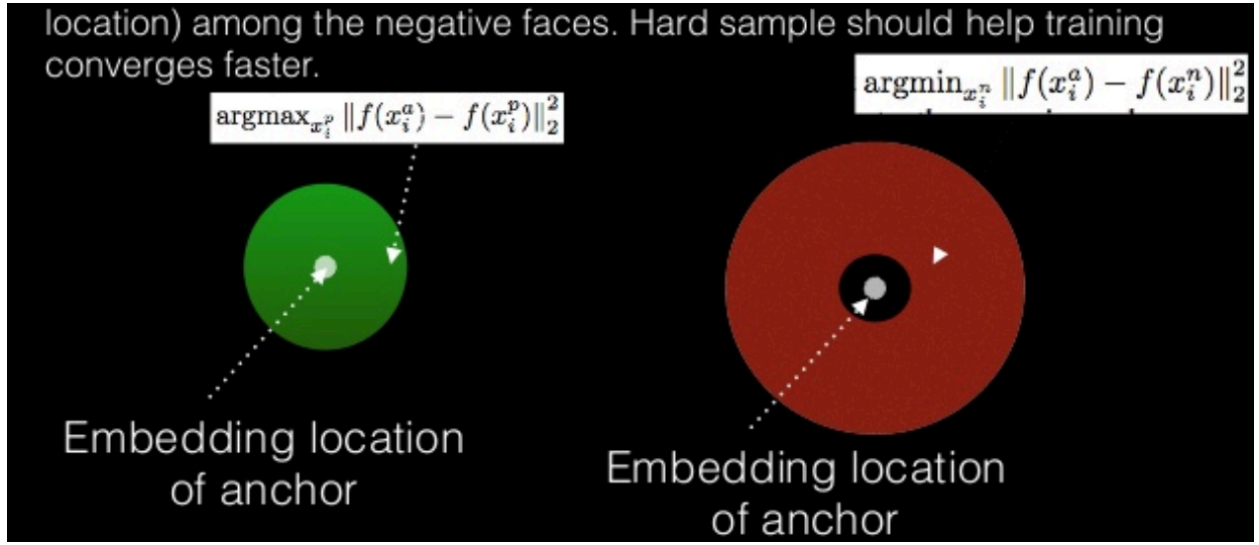
$$\forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in \mathcal{T} \,.$$

Train하는 동안, A,P,N 이 랜덤으로 골라진다면, 위의 조건을 너무 쉽게 만족.
그렇게 되면 학습이 잘 되지 않는다!!!

Train하기 어려운 **hard**한 triplet을 고르자.

3.2 Triplet Selection

- Hard positive : $\text{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$ a
- Hard negative : $\text{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$.



3.2 Triplet Selection

- Offline

- neg_distance – pos_distance < margin인 데이터가 hard triplets
- but, 매 step마다 모든 데이터를 보는 것은 너무 많다.

- Online

- Mini batch 당, 한 ID당 40개의 데이터 사용
- 모든 positive 사용 -> 안정적인 학습, 초반에 약간 빠른 수렴
- 학습 초반에 Hard-negative를 사용하는 것은 불안정 -> semi-hard 사용

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 \ .$$

3.3 Method - Models

- Category 1
- Zeiler&Fergus 기반 모델
- 1x1 convolution 추가
- NN1

- Category 2
- GoogleNet 기반 모델
- NN2 ~ 4의 입력크기: 220x220, 160x160, 96x96
- NNS1~4 : 모바일을 위한 작은 모델들

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Table 1. **NN1.**

Datasets and Evaluation

- Datasets

- Train datasets : 비공개
- Test datasets : LFW, YouTube Faces Database

- Evaluation

- Hold out cross validation 사용
- Validation rate: 같은 사람을 같다고 예측한 비율
- False accept rate: 다른 사람을 같다고 예측한 비율

```
tpr = 0 if (tp+fn==0) else float(tp) / float(tp+fn)
fpr = 0 if (fp+tn==0) else float(fp) / float(fp+tn)
acc = float(tp+tn)/dist.size
return tpr, fpr, acc
```

Experiments

- Image Quality

jpeg q	val-rate
10	67.3%
20	81.4%
30	83.9%
50	85.5%
70	86.1%
90	86.5%

#pixels	val-rate
1,600	37.8%
6,400	79.5%
14,400	84.5%
25,600	85.7%
65,536	86.4%

- Training data size

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

- Embedding Dimensionality

#dims	VAL
64	$86.8\% \pm 1.7$
128	$87.9\% \pm 1.9$
256	$87.7\% \pm 1.9$
512	$85.6\% \pm 2.0$

Figure 7. **Face Clustering.** Shown is an exemplar cluster for one user. All these images in the users personal photo collection were clustered together.

Summary

1. Face verification을 위해 직접 Euclidean space로 embedding 하는 것을 학습

2. 최소한의 alignment만 필요 (tight한 crop과정만 필요)

- References

- Wang, Mei, and Weihong Deng. "Deep face recognition: A survey." *arXiv preprint arXiv:1804.06655* (2018).

- Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

- Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 201

- Sun, Yi, et al. "Deep learning face representation by joint identification-verification." *Advances in neural information processing systems*. 2014.

- 감사합니다 -