

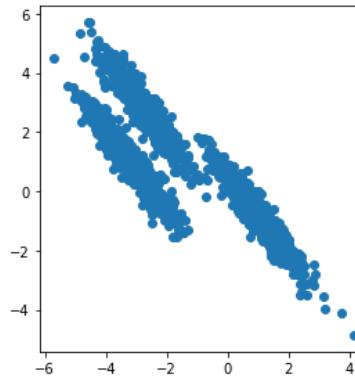


Clustering Algorithms and their Application to Facial Image Analysis

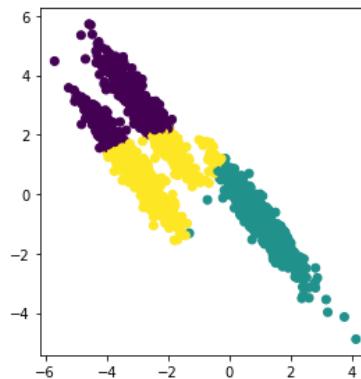
Hamid Sadeghi

nextera, FaceCup 1400 (2022)

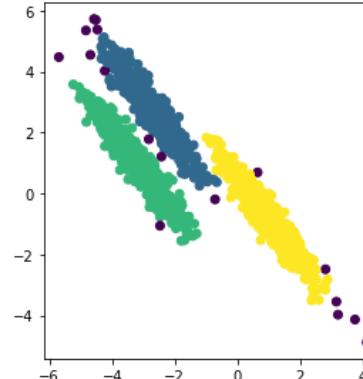
Clustering (summary)



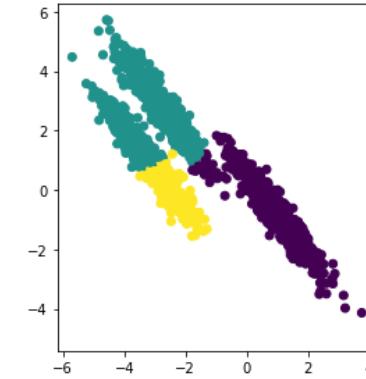
K-means



DBSCAN



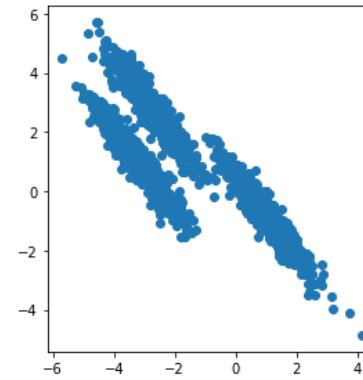
Agglomerative



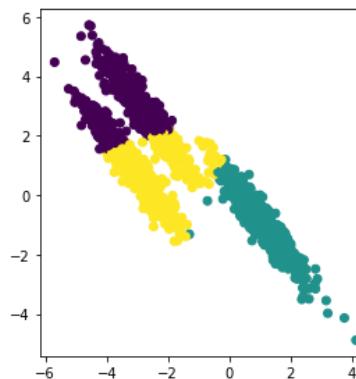
Clustering (summary)

External Evaluation (supervised):

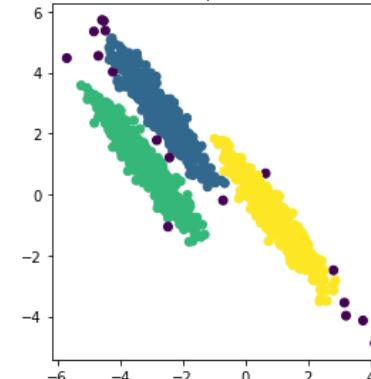
- ✓ Purity
- ✓ Rand Index
- ✓ NMI
- ✓ F-measure



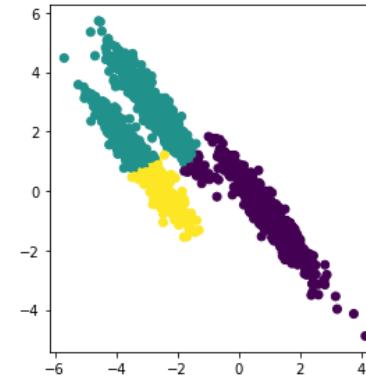
K-means



DBSCAN

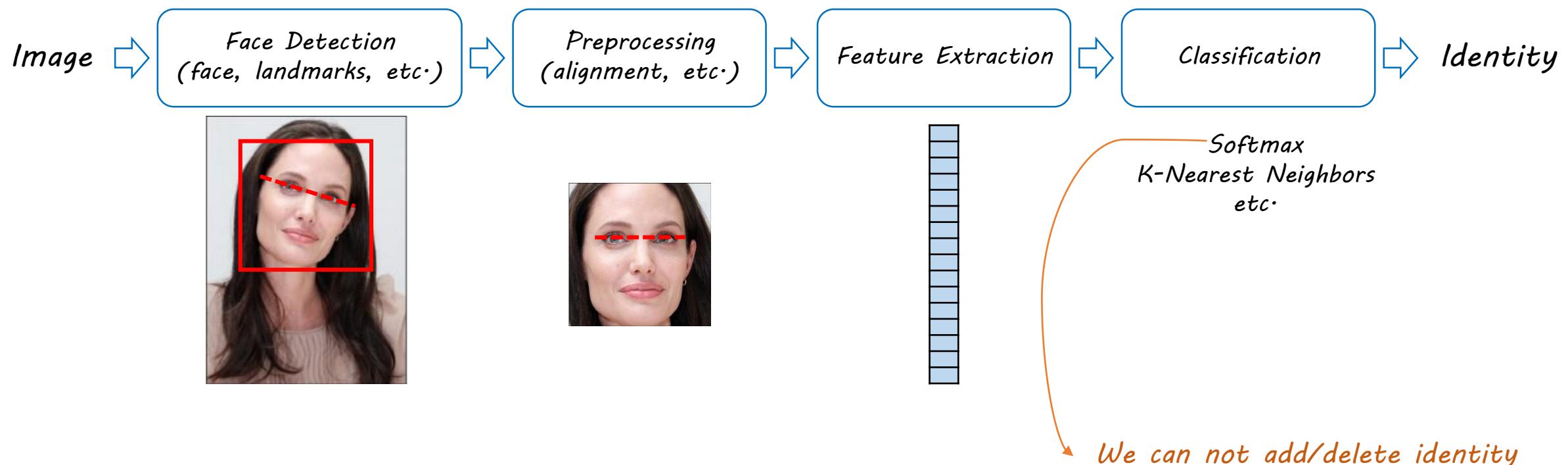


Agglomerative



Face Analysis (summary)

- Block diagram of a face recognition system



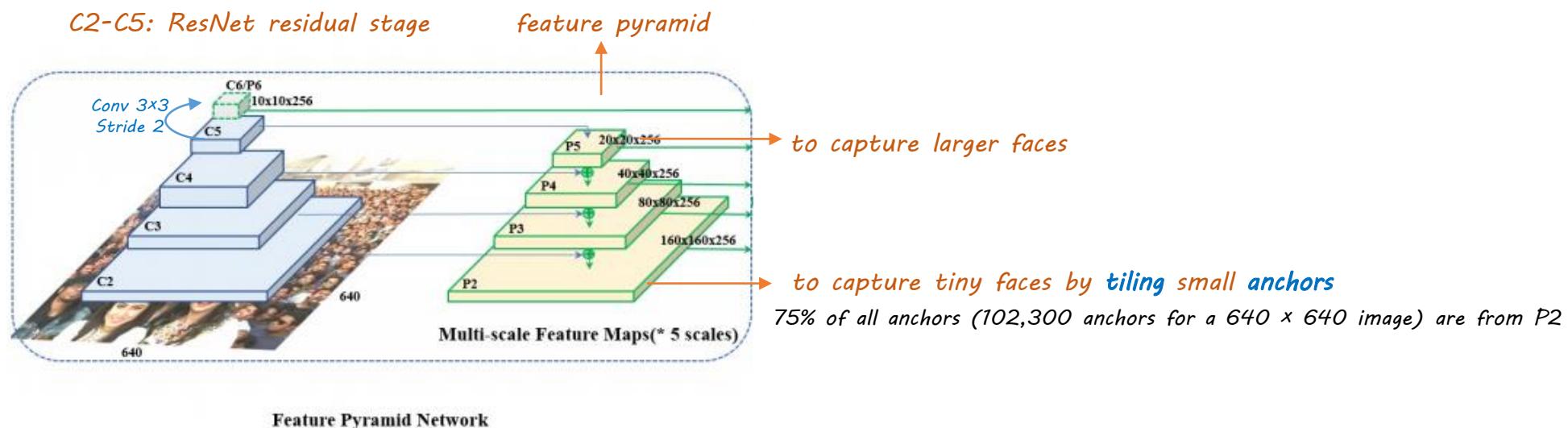


Contents

- ✓ *Introduction*
- ✓ *Clustering Algorithms*
- ✓ *Evaluation*
- *Face Analysis*
 - ✓ *Introduction*
 - ❑ *Face Detection & Preprocessing*
 - ❑ *Face Recognition*
 - ❑ *A Complete Face Clustering Algorithm*

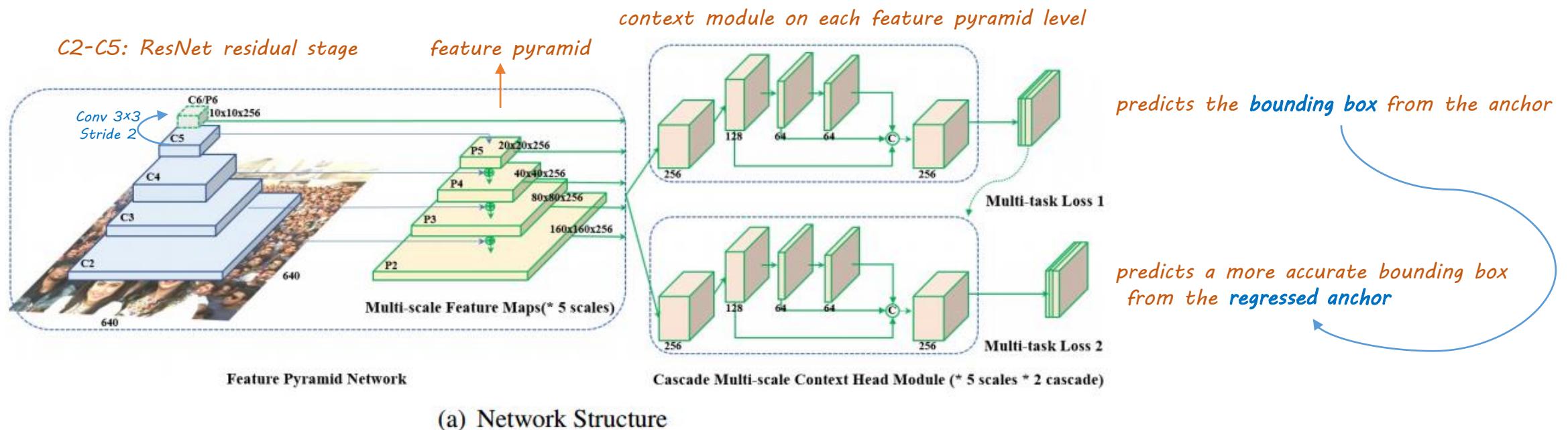
Face Analysis: Face Detection

- RetinaFace



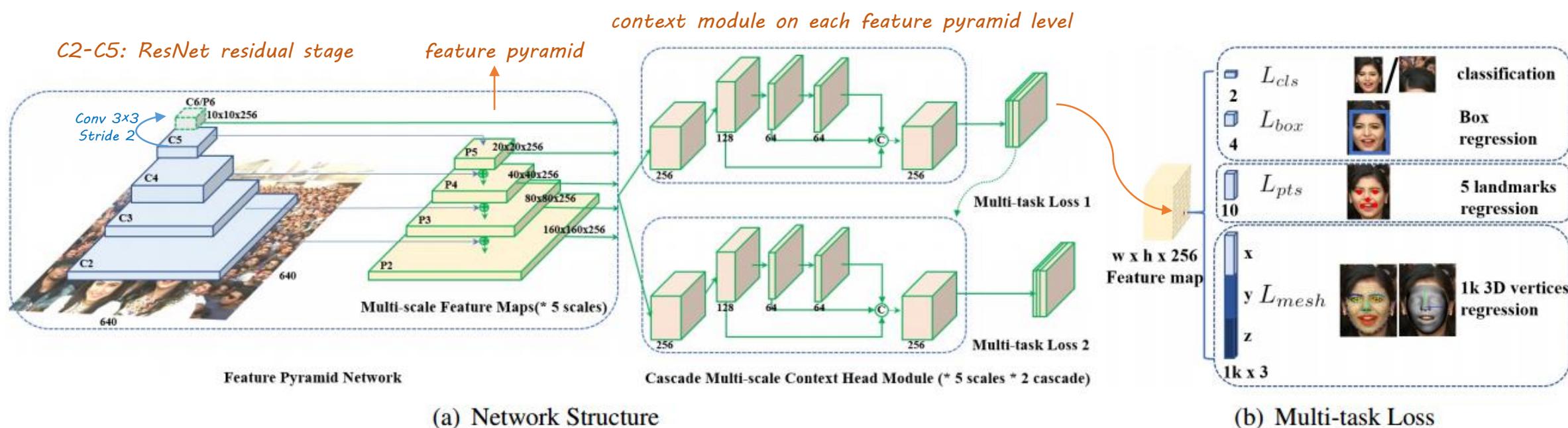
Face Analysis: Face Detection

- RetinaFace



Face Analysis: Face Detection

- RetinaFace

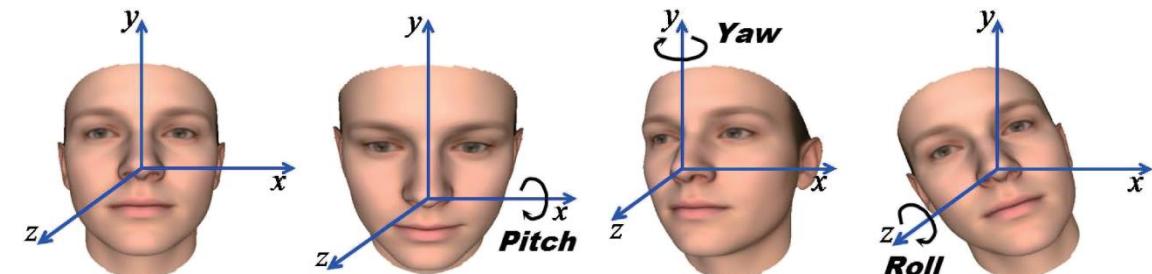


Code:

<https://github.com/deepinsight/insightface/tree/master/detection/retinafece>

Face Analysis: Preprocessing

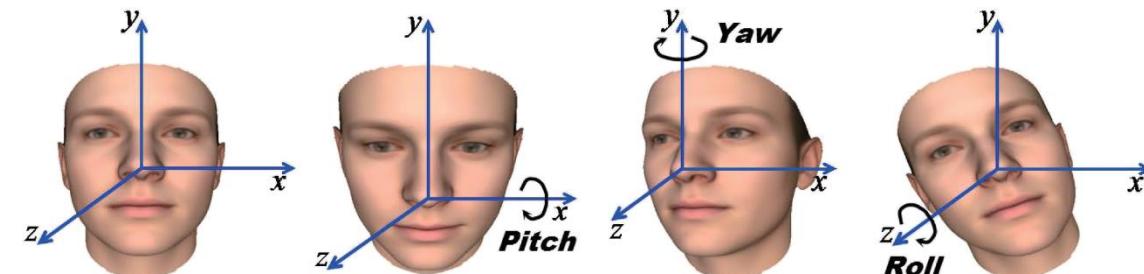
- Head pose



G. Sang, et al., Learning toward practical head pose estimation, SPIE, 2017

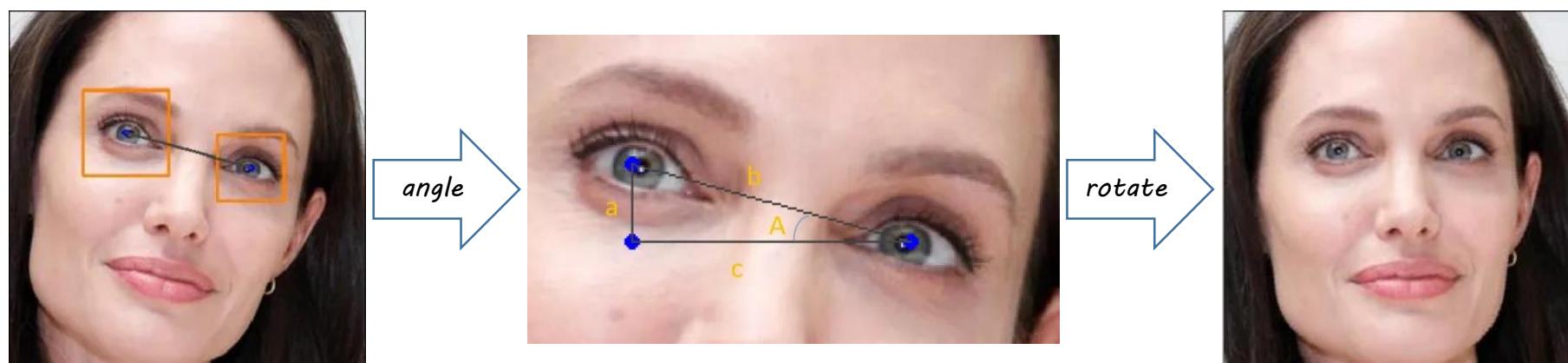
Face Analysis: Preprocessing

- Head pose



G. Sang, et al., Learning toward practical head pose estimation, SPIE, 2017

- Roll:

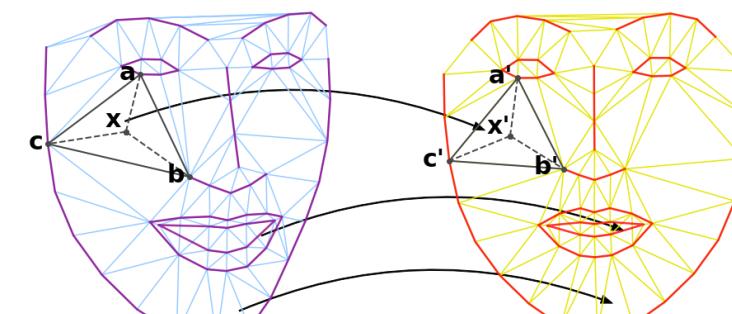


Code: <https://sefiks.com/2020/02/23/face-alignment-for-face-recognition-in-python-within-opencv/>

Face Analysis: Preprocessing

- Pitch and Yaw

piecewise affine warping

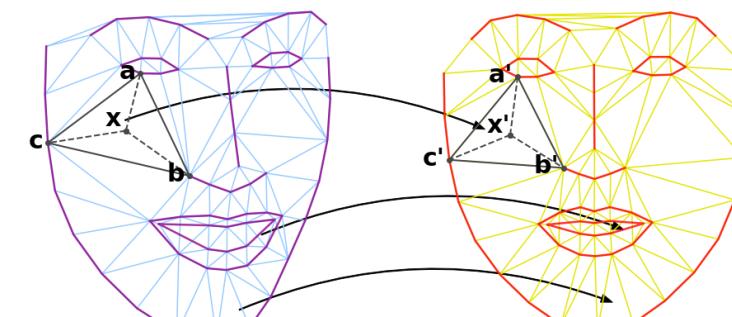


Delaunay triangulation

Face Analysis: Preprocessing

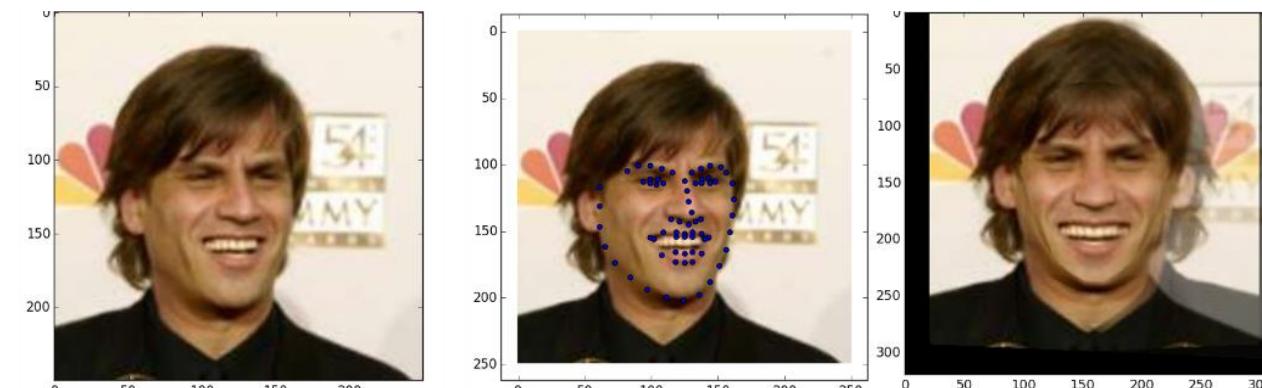
- Pitch and Yaw

piecewise affine warping



<http://jeankossaifi.com/pages/gagan.html>

Face Frontalization

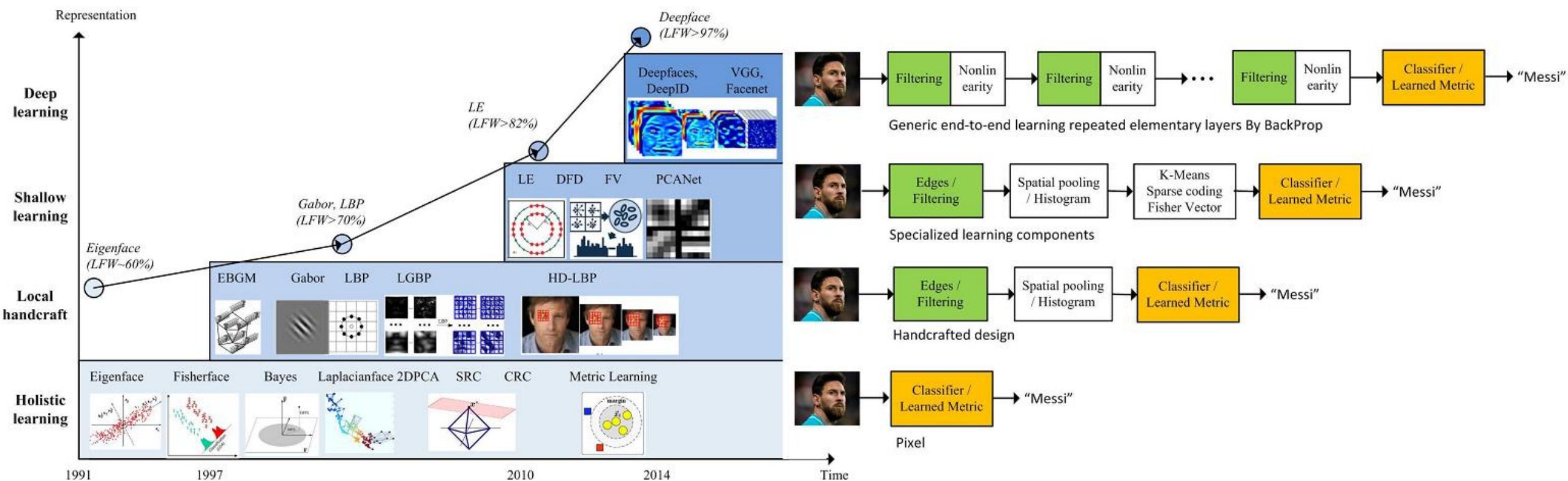


Code: <https://github.com/dougsouza/face-frontalization>

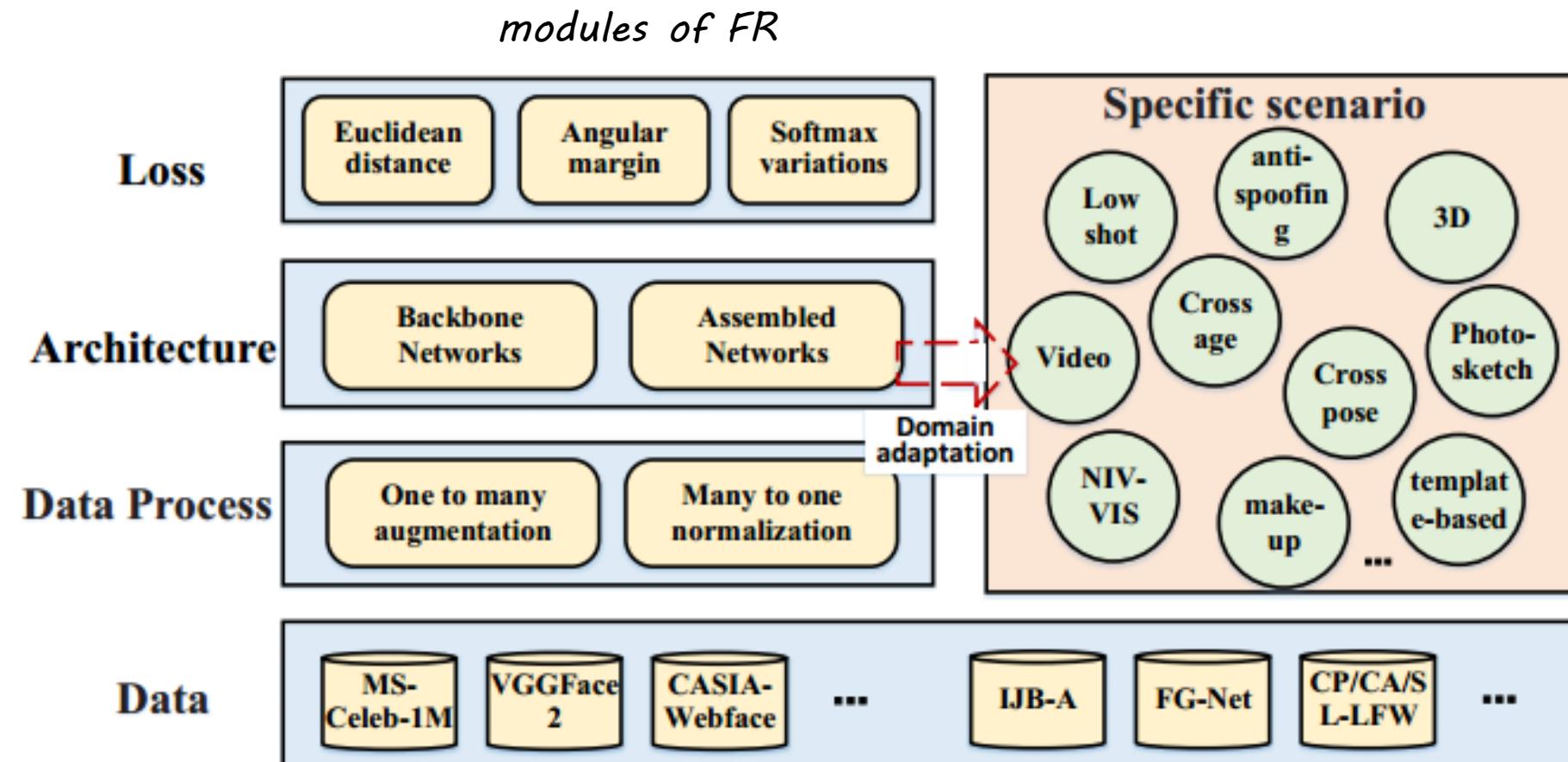
Insightface Code: <https://github.com/deepinsight/insightface/tree/48cc12e3d83c5e98122068eaf04f5d9829c8b399/alignment/coordinateReg>

Face Analysis: Face Recognition

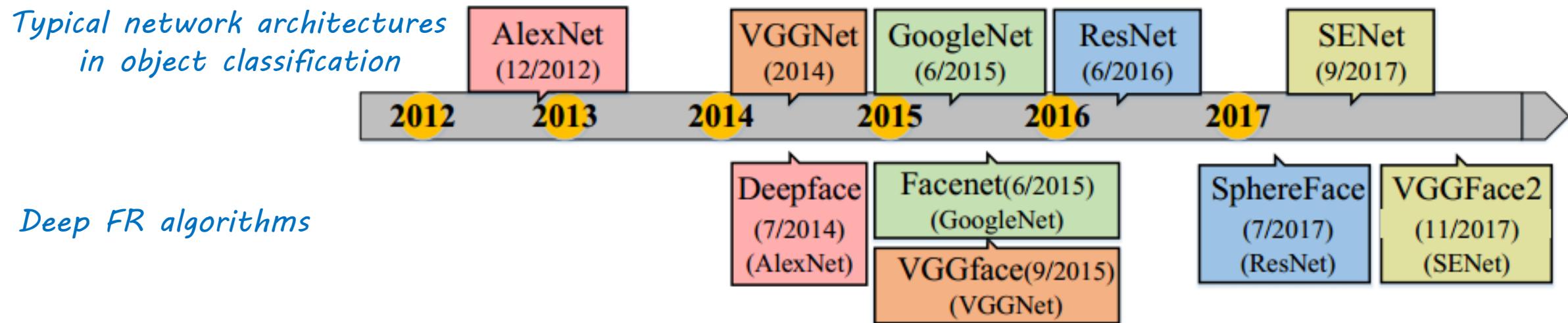
Milestones of face representation for recognition



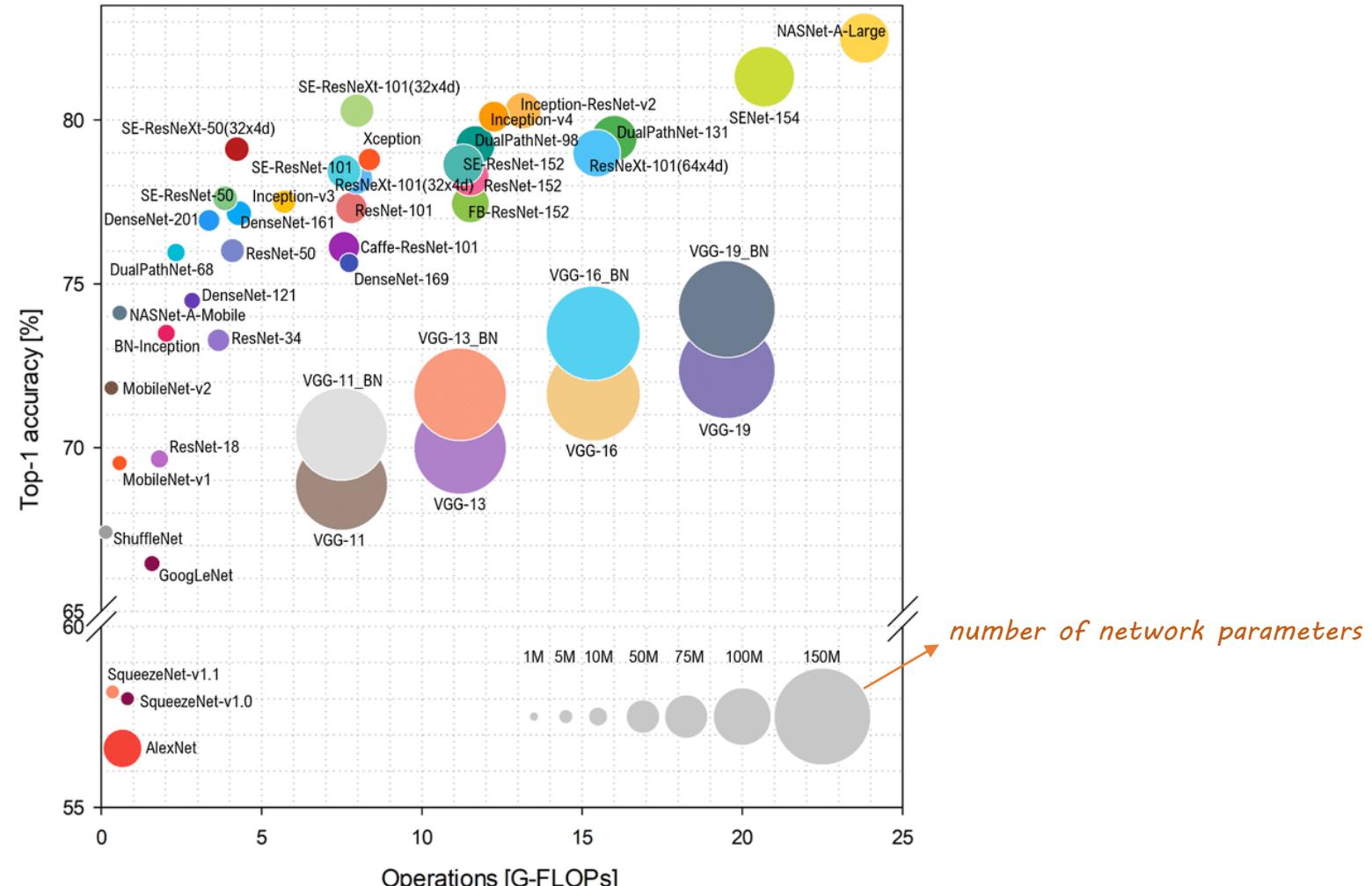
Face Analysis: Face Recognition



Face Analysis: Face Recognition



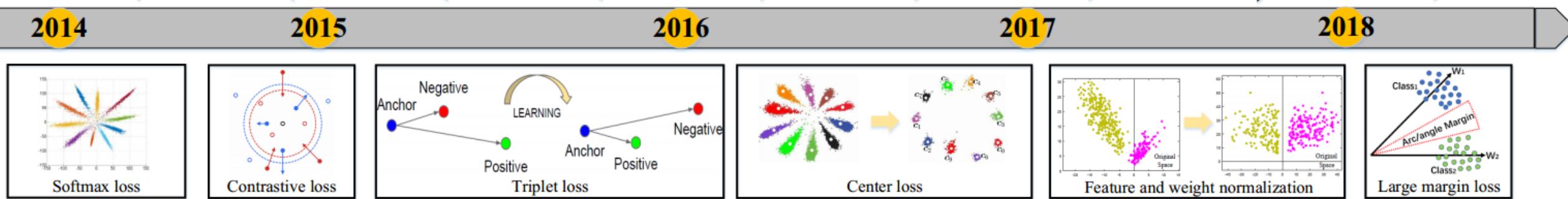
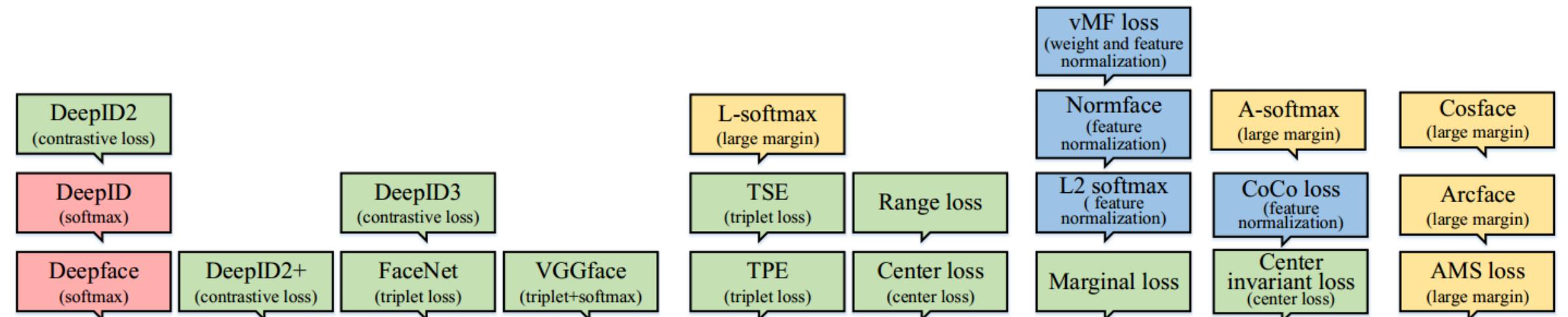
Face Analysis: Face Recognition



A. Canziani, An analysis of deep neural network models for practical applications, 2016
S. Bianco, Benchmark Analysis of Representative Deep Neural Network Architectures, 2018

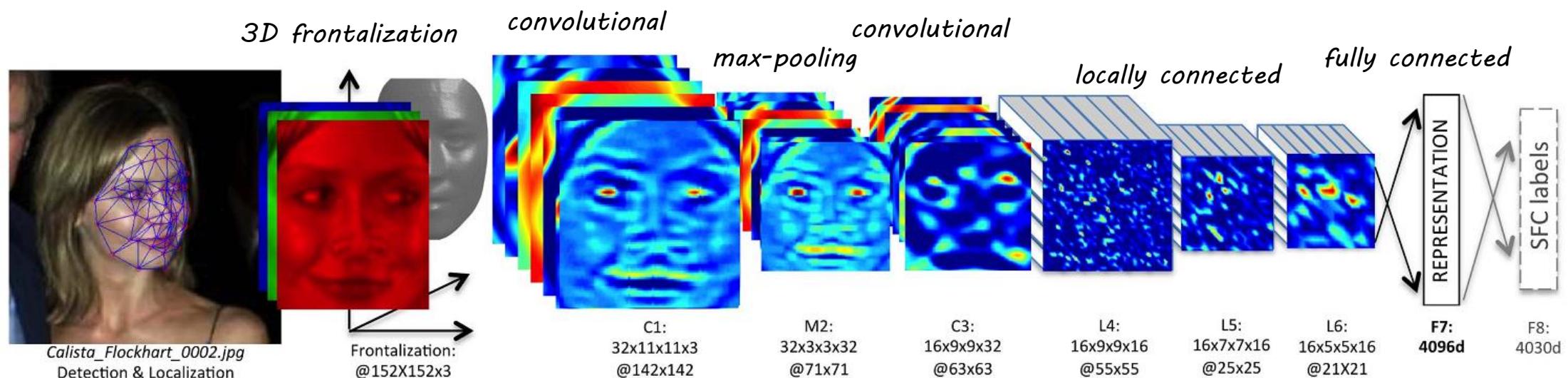
Face Analysis: Face Recognition

Loss function



Face Analysis: Face Recognition

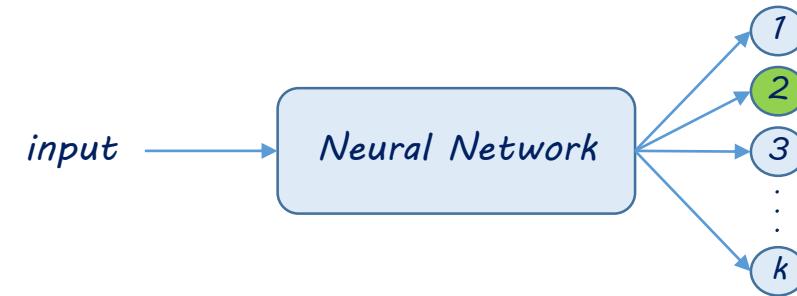
- DeepFace



- 1 - Train the NN using Softmax (for multi-class classification or identification)
- 2 - Train the two topmost layers using Siamese (for verification)

Face Analysis: Face Recognition

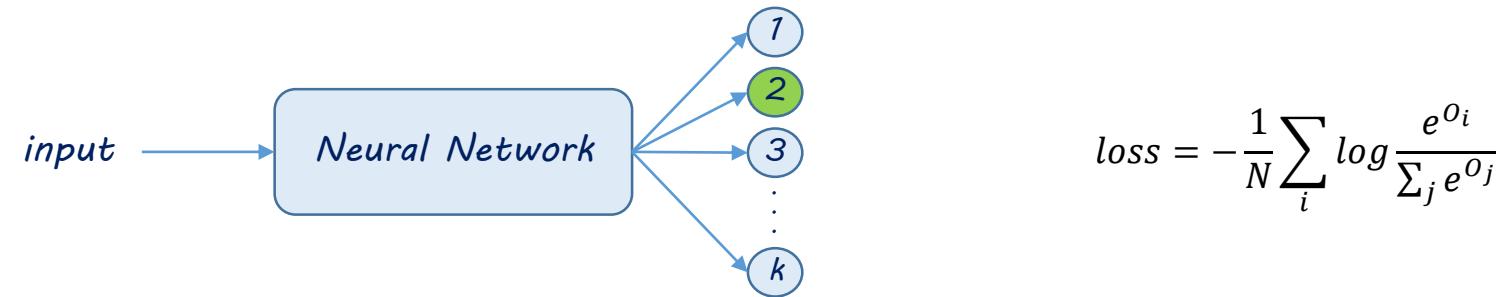
- Softmax



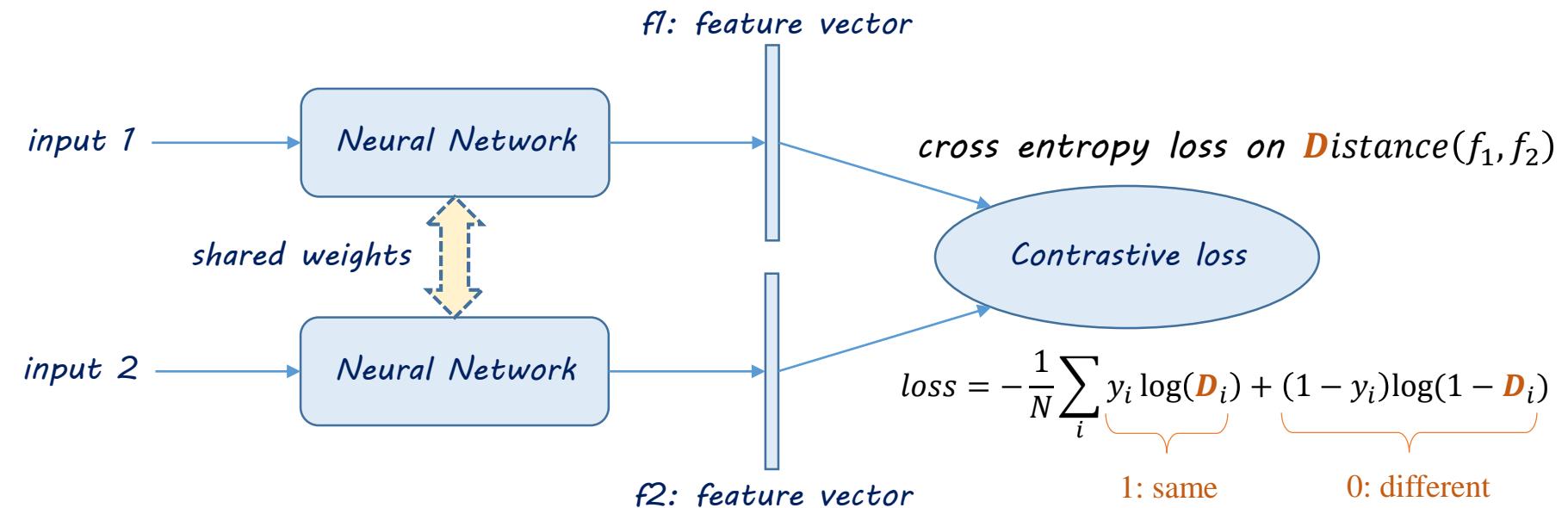
$$\text{loss} = -\frac{1}{N} \sum_i \log \frac{e^{o_i}}{\sum_j e^{o_j}}$$

Face Analysis: Face Recognition

- **Softmax**



- **Siamese**



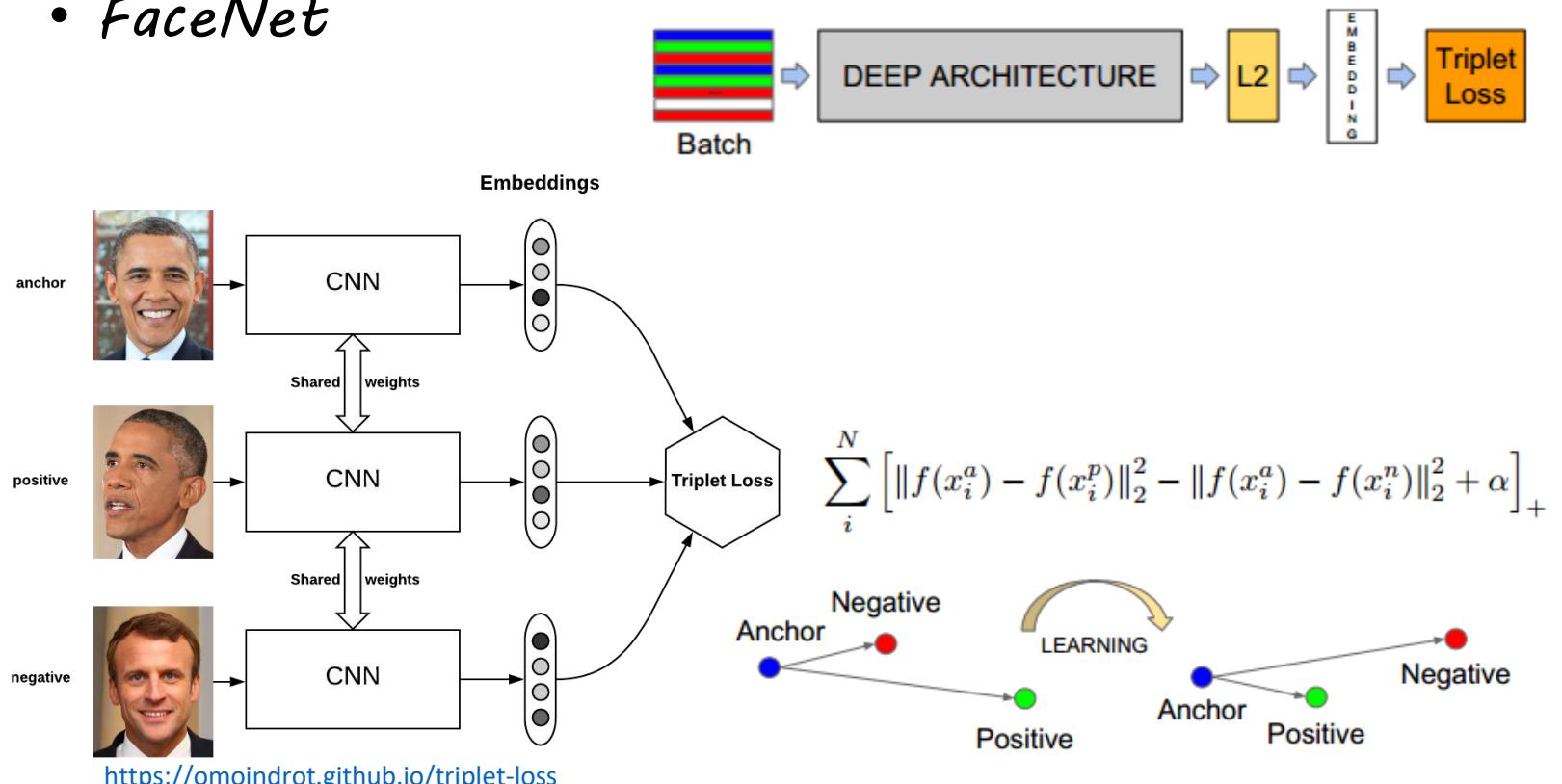
Face Analysis: Face Recognition

- FaceNet



Face Analysis: Face Recognition

- *FaceNet*



Code: <https://github.com/davidsandberg/facenet>

Face Analysis: Face Recognition

- ArcFace

Softmax:

$$L_1 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}}$$

Face Analysis: Face Recognition

- ArcFace

Softmax:

$$L_1 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}}$$

Annotations:

- #classes: points to the term n in the denominator.
- Batch size: points to the term N in the denominator.
- Label of x_i : points to the term y_i in the exponent.
- i^{th} data: points to the term x_i in the exponent.
- bias: points to the term b_{y_i} .

Deng, et al.: “Softmax loss function does not explicitly optimise the feature embedding to enforce higher similarity for intra-class samples and diversity for inter-class samples”

Pose, Age, etc.

Let $b = 0$, $\|W_j\| = 1$, $\|x_i\| = s$
 $W_j^T x_i = \|W_j\| \|x_i\| \cos \theta_j$

$$\rightarrow L_2 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

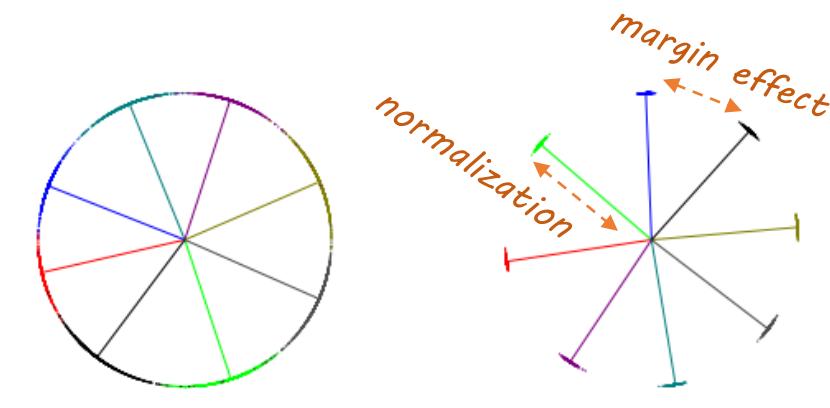
$$L_3 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s (\cos(\theta_{y_i} + m))}}{e^{s (\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}$$

margin

Face Analysis: Face Recognition

- ArcFace

8 identities with 2D normalized features:



(a) Softmax

(b) ArcFace

- Code:

<https://github.com/deepinsight/insightface>

python library, Originally MXNet, newly PyTorch

Face Analysis: Face Recognition

- *Implementations:*

- *face recognition using dlib:*

https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/face_recognition_using_dlib.ipynb

- *face recognition using ArcFace (insightface):*

<https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/arcface.ipynb>



A Complete Face Clustering Algorithm



Face Analysis: Face Clustering

- *Implementations:*
 - *Face clustering using ArcFace on a small set:*
https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/face_clustering_arcface.ipynb
 - *Face clustering using ArcFace on FaceCup sample dataset:*
https://colab.research.google.com/github/hamidsadeghi68/face-clustering/blob/main/face_clustering_arcface_facecup_samples.ipynb

