CSE 152B: Computer Vision II

Manmohan Chandraker

Lecture 9: Face Recognition

Course details

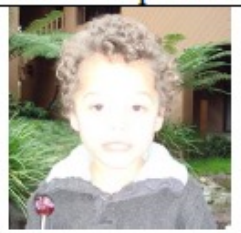
- Class webpage:
  - http://cseweb.ucsd.edu/~mkchandraker/classes/CSE152B/Spring2025/
- Instructor email:
  - mkchandraker@ucsd.edu
- TA: Mustafa Yaldiz
  - Emails: myaldiz@ucsd.edu
- Grading
  - 40% assignments
  - 25% midterm (open notes)
  - 35% final exam (open notes)
- Aim is to learn together, discuss and have fun!

Overall goals for the course

- Introduce the state-of-the-art in computer vision
- Study principles that make them possible
- Get understanding of tools that drive computer vision
- Enable one or all of several such outcomes
  - Pursue higher studies in computer vision
  - Join industry to do cutting-edge work in AI
  - Gain an appreciation of modern AI technologies

Recap

Unconstrained Face Recognition

Scenario	External occlusion	Self occlusion	Facial accessories	Limited field of view (FOV)	Extreme illumination	Sensor saturation
Examples	occlusion by other objects	non-frontal pose	hat, sunglasses, scarf, mask	partially out of camera's FOV	gloomy or highlighted facial area	underexposure or overexposure
Image						

[Liao et al., Partial Face Recognition, PAMI 2013]

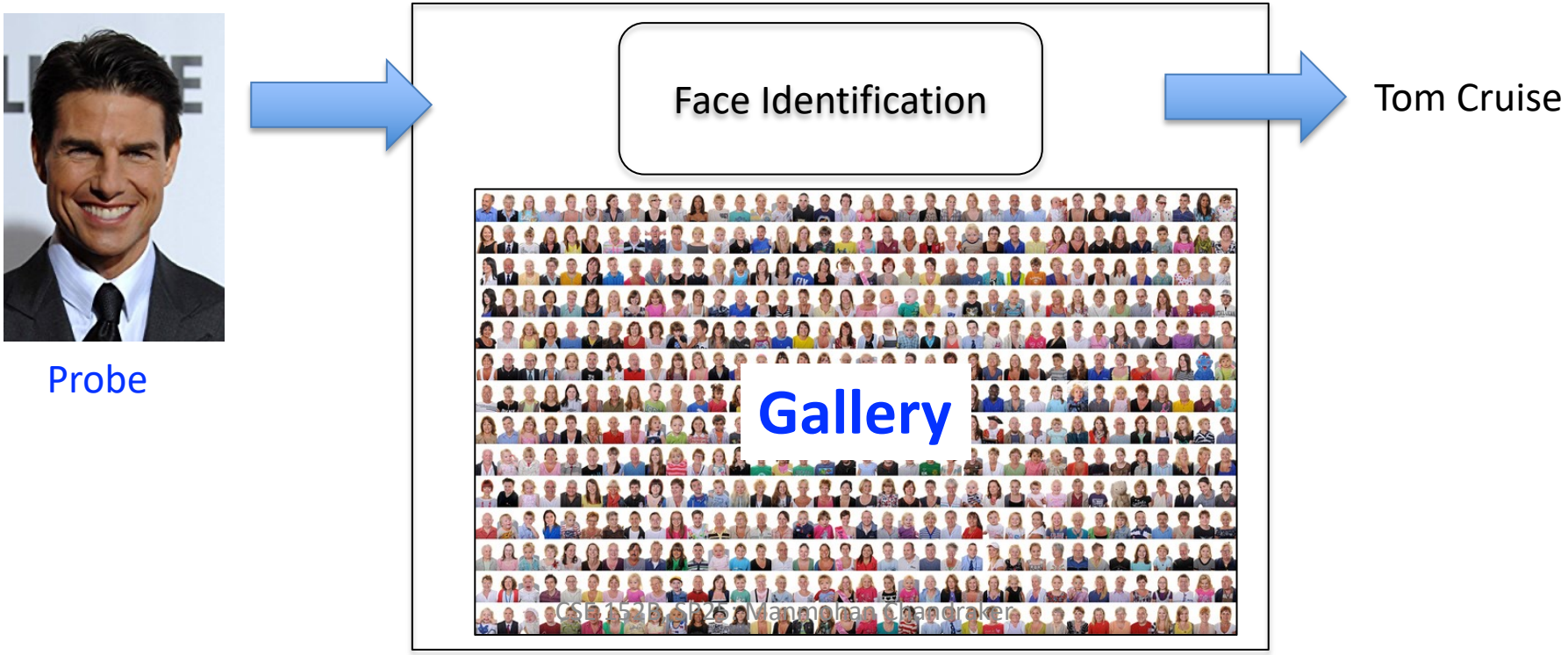
Face Recognition on LFW Benchmark

- Human performance : **99.20%**
- Local Binary Patterns : 95.17%
- DeepFace : 97.35 %
- DeepID2 : 99.15%
- FaceNet : **99.63%**

Face Identification

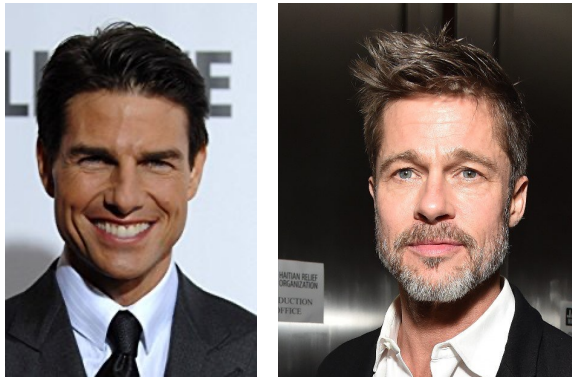
- Closed set identification: assign one of gallery identities to probe image
- Galleries can be very large, high chance of similar appearances

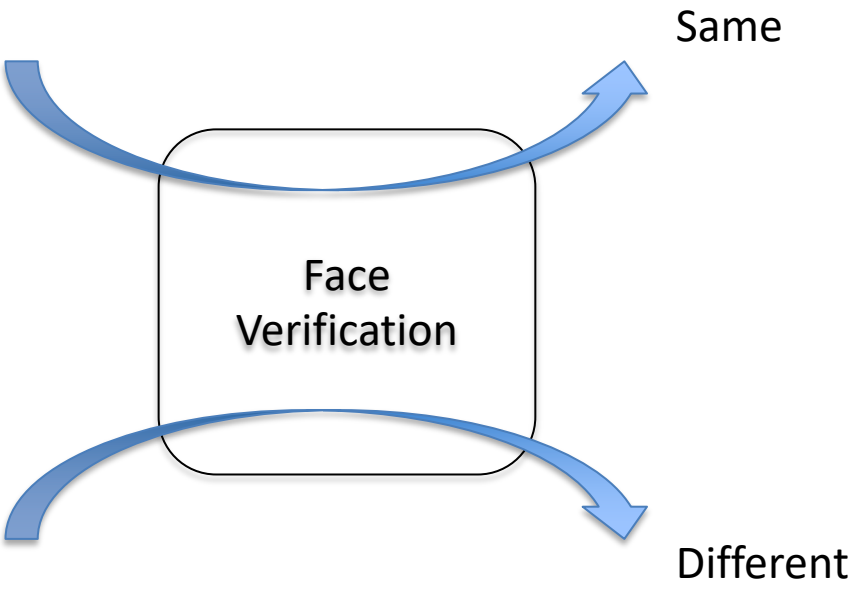
- Goal is to have sharp decision boundary between gallery identities
- Feature need not generalize to other tasks (identities outside the gallery)



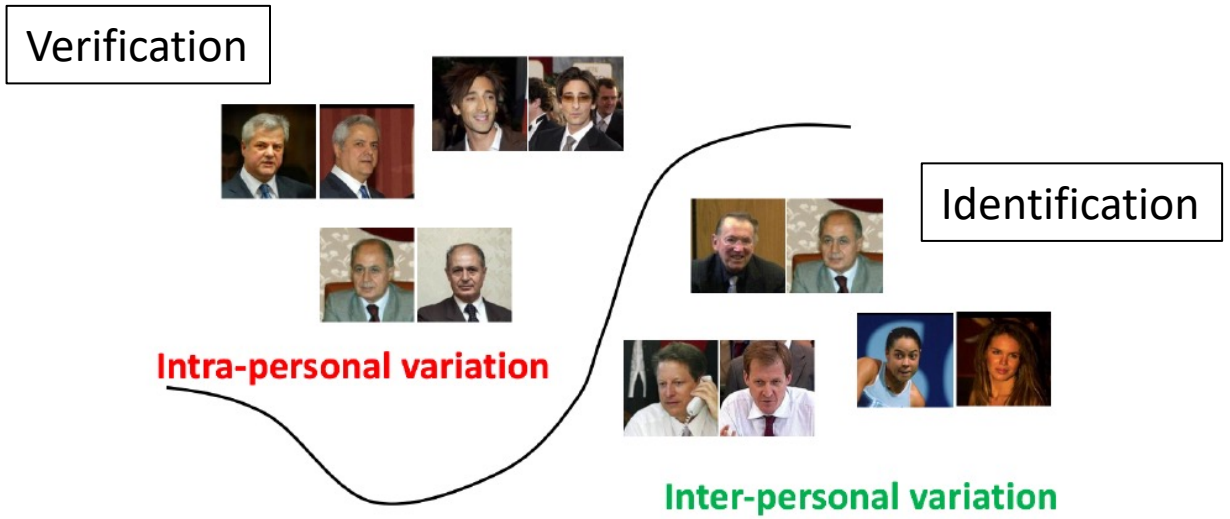
Face Verification

- Given a pair of face images:
  - A squared L2 distance $D(x_i, x_j)$ is used to determine same or different
  - Good embedding: true matches will lie within a small value of $D(x_i, x_j)$





Verification and Identification Signals



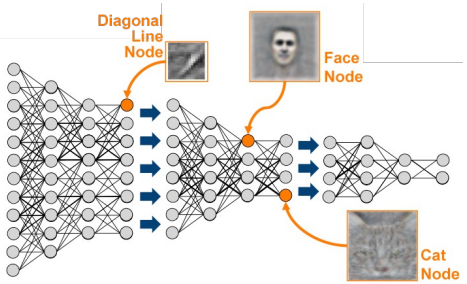
- **Identification:**
  - Distinguish images of one identity from another identity
  - Favors large distance between clusters
  - Stronger learning signal, but need not generalize to new identities
- **Verification:**
  - Match two images of an individual across large appearance variations
  - Favors tight clusters for each identity
  - Weaker learning signal, but feature applicable to new identities

Steps in Face Recognition

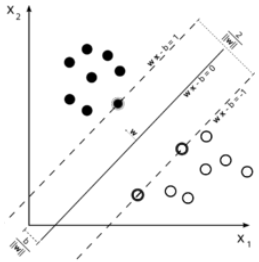
- Face Detection
  - Localize the face

- Face Alignment
  - Factor out 3D transformation

- Feature Extraction
  - Find compact representation



- Classification
  - Answer the question



Challenges in Face Alignment

- Infer 3D from 2D
  - Slight occlusion
  - Lighting condition
  - Head orientation
  - Non rigid deformation

DeepFace Alignment: Substep 1

- 2D feature point extraction
- 2D alignment $x_{anchor} = (S * R * T)x_{source}$
- Only for in plane alignment

Fiducial Point Detection

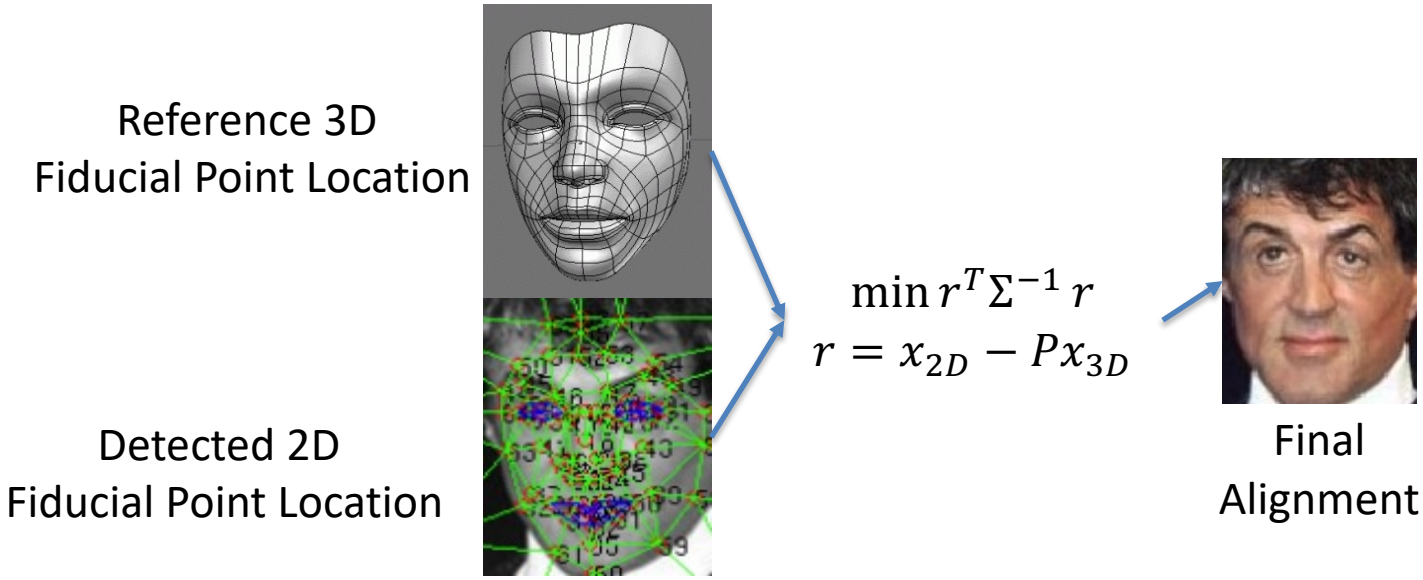
2D Transformation

Until convergence

[Taigman et al., DeepFace]

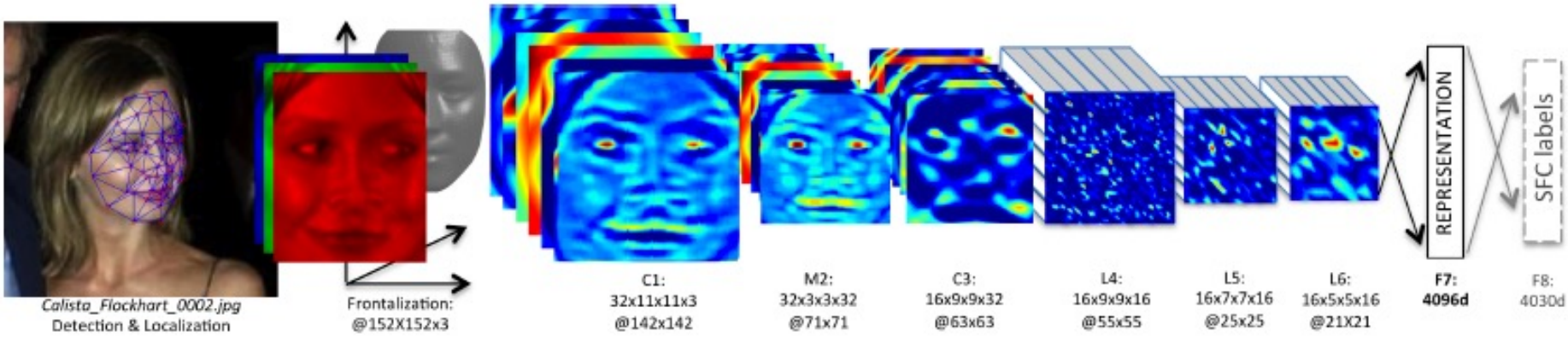
DeepFace Alignment: Substep 2

- 3D feature point extraction
- 3D alignment



[Taigman et al., DeepFace]

Architecture

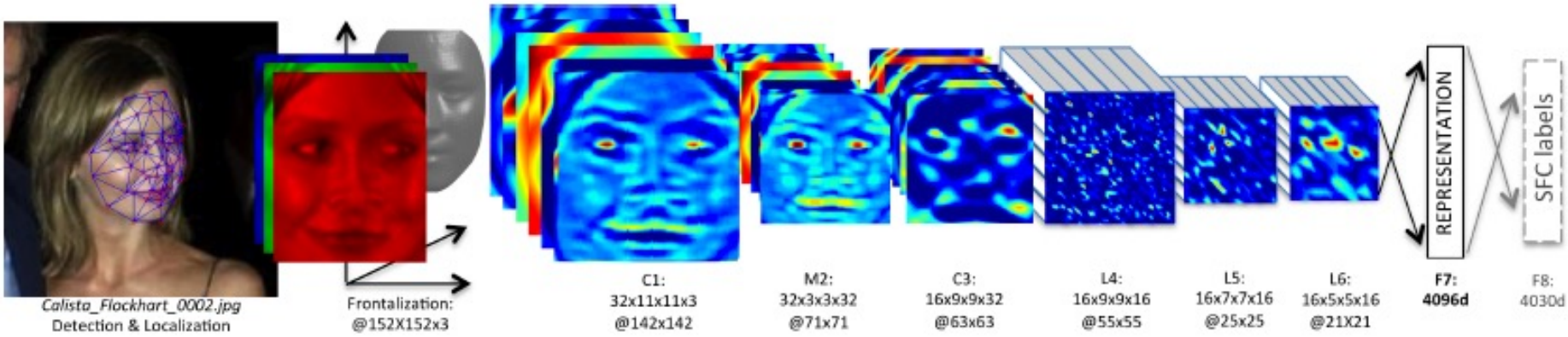


Layer 1-3 : Intuition

- Convolution layers - extract low-level features (e.g. simple edges and texture)

[Taigman et al., DeepFace]

Architecture



Layer 1-3 : Intuition

- Convolution layers - extract low-level features (e.g. simple edges and texture)

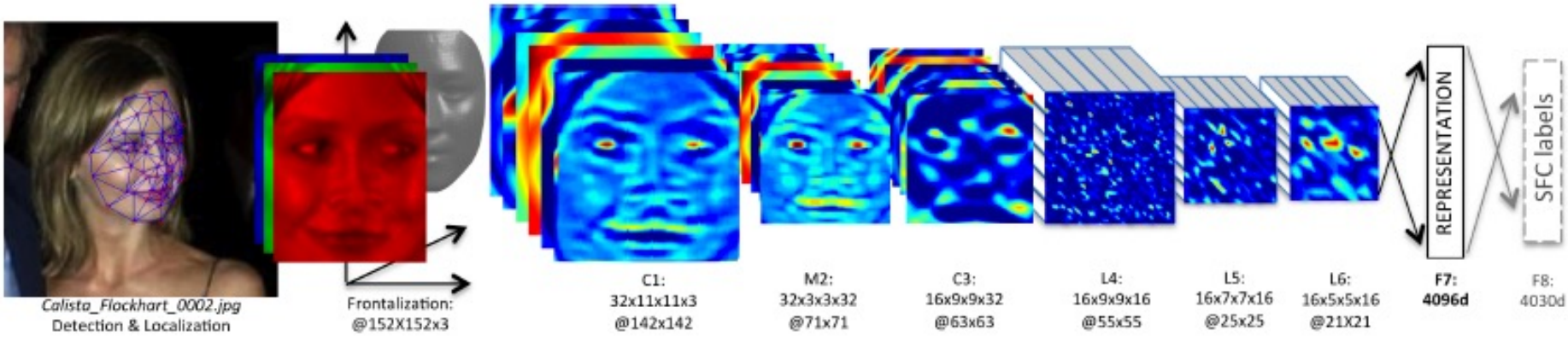
Layer 4-6: Intuition

- Apply filters to different locations on the map
- Similar to a conv. layer but spatially dependent

- Different regions of an aligned image have different local statistics
  - Aligned images with similar semantic concepts are being considered
  - A large training dataset is available, can handle increased parameters

[Taigman et al., DeepFace]

Architecture



- Layer F7 is fully connected and generates 4096d vector
- Sparse representation of face descriptor
- Layer F8 is fully connected and generates 4030d vector
- F8 calculates probability with softmax $p_k = \frac{\exp(o_k)}{\sum_h \exp(o_h)}$
- Cross-entropy loss function: $\sum_{i=1}^{n} -p_i \log \hat{p}_i$

Target probability distribution
p_i = 1 for class t and 0 for other i

Predicted probability distribution

[Taigman et al., DeepFace]

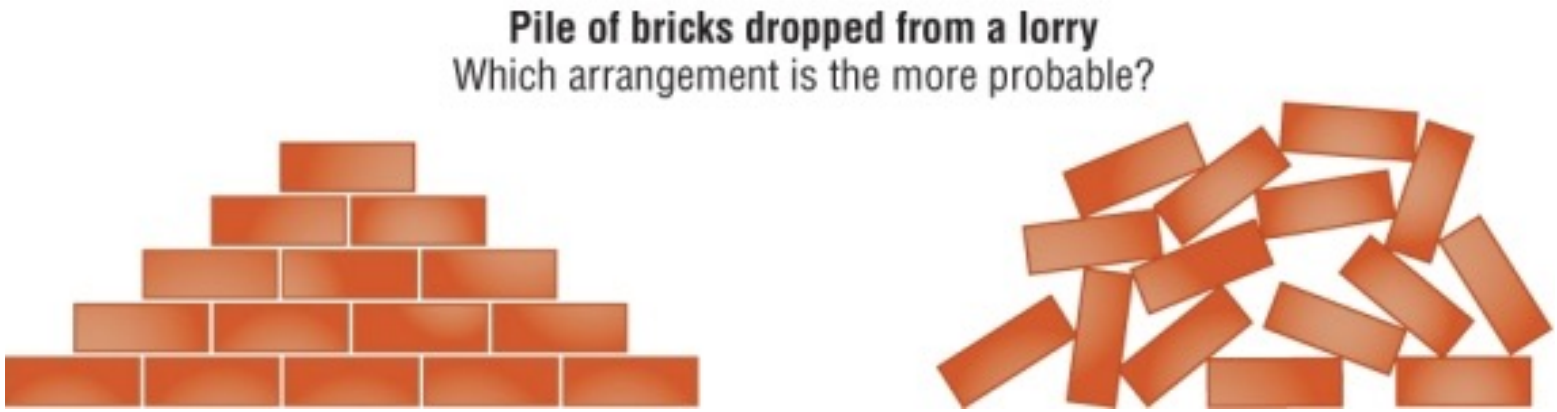
Entropy

- A measure of randomness in a system
  - The higher the entropy, the less ordered is the system

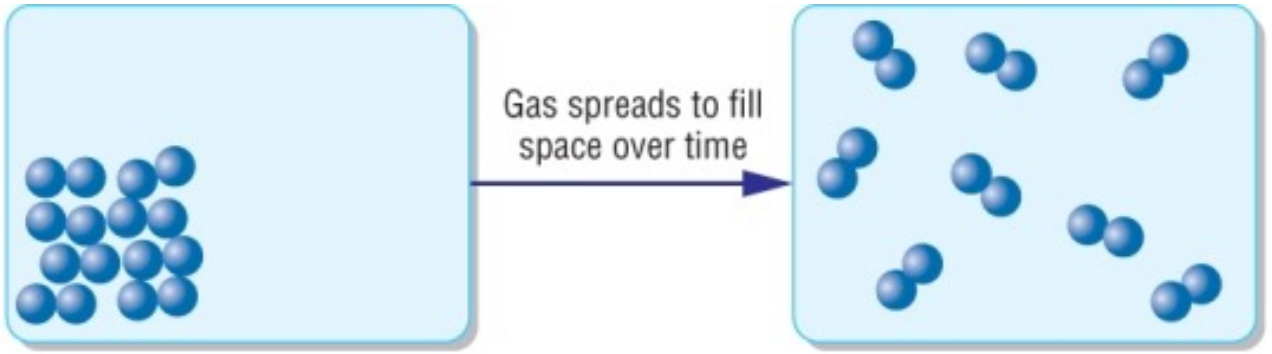
Pile of bricks dropped from a lorry
Which arrangement is the more probable?

Entropy

- A measure of randomness in a system
  - The higher the entropy, the less ordered is the system



  - Natural systems tend to assume a state of higher entropy

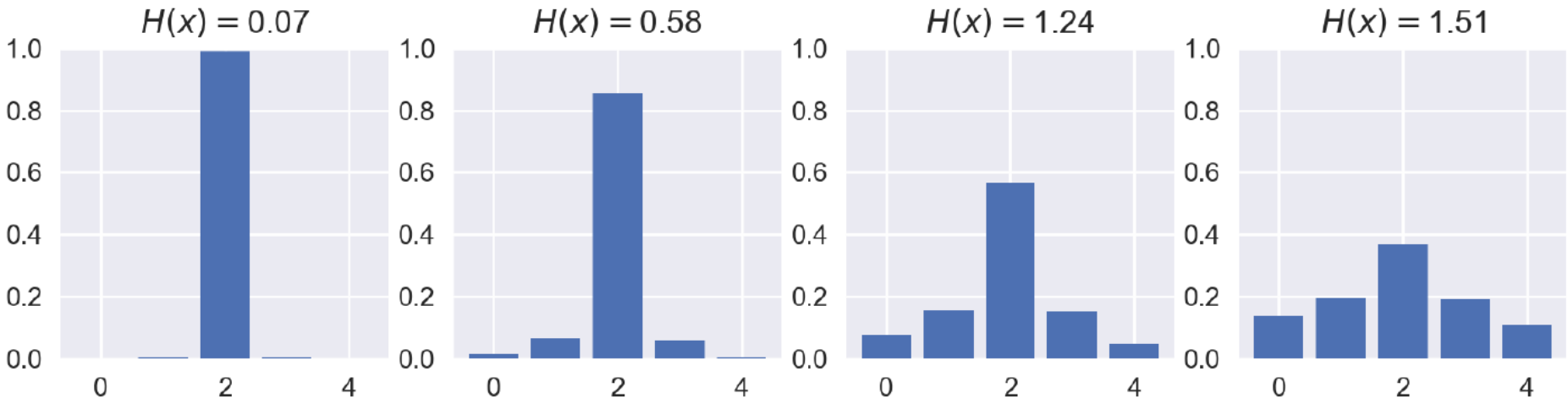


Entropy in Classification

- Suppose we are doing 5-way classification
- Some of these output distributions are better than others

Entropy in Classification

- Suppose we are doing 5-way classification
- Some of these output distributions are better than others



- More peaky distribution is better for classification

- More peaky distribution = Lower entropy
- More uncertainty = Higher entropy

Information Content

- Consider an unfair coin with p(Heads) = 0.99
  - A coin toss that yields H is not surprising
  - But a toss that yields T is very surprising
- Information content of a stochastic event E

$$I(E) = -log[Pr(E)] = -log(P)$$

  - Logarithm in base 2: information in bits
  - Natural logarithm: information in nats
- For unfair coin,
  - Information in event Heads: –log(0.99) = 0.01 bits
  - Information in event Tails: -log(0.01) = 6.64 bits
  - Matches intuition for Tails being a more surprising event than Heads

Entropy

- Entropy: expected rate of information from stochastic process
  - For a random variable X, expected value is

$$E[X] = \sum_{i=1}^{n} x_i p_i$$

  - When the random variable is information, expectation is

$$H(X) = E[I(X)] = E[-log(P(X))] = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

  - For the unfair coin with p(H) = 0.99, we have

$$H(X) = -(0.99log(0.99) + 0.01log(0.01)) = 0.08bits$$

Entropy

- Entropy: expected rate of information from stochastic process
  - For a random variable X, expected value is

$$E[X] = \sum_{i=1}^{n} x_i p_i$$

  - When the random variable is information, expectation is

$$H(X) = E[I(X)] = E[-log(P(X))] = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

  - For the unfair coin with p(H) = 0.99, we have

$$H(X) = -(0.99log(0.99) + 0.01log(0.01)) = 0.08bits$$

  - For a fair coin with p(Heads) = 0.5, we have

$$H(X) = -(0.5\ log(0.5) + 0.5\ log(0.5)) = 1\ bit$$

Entropy

- Entropy: expected rate of information from stochastic process
  - For a random variable X, expected value is

$$E[X] = \sum_{i=1}^{n} x_i p_i$$

  - When the random variable is information, expectation is

$$H(X) = E[I(X)] = E[-log(P(X))] = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

  - For the unfair coin with p(H) = 0.99, we have

$$H(X) = -(0.99log(0.99) + 0.01log(0.01)) = 0.08bits$$

  - For a fair coin with p(Heads) = 0.5, we have

$$H(X) = -(0.5\ log(0.5) + 0.5\ log(0.5)) = 1\ bit$$

- _____ uncertainty = _____ entropy

Entropy

- Entropy: expected rate of information from stochastic process
  - For a random variable X, expected value is

$$E[X] = \sum_{i=1}^{n} x_i p_i$$

  - When the random variable is information, expectation is

$$H(X) = E[I(X)] = E[-log(P(X))] = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

  - For the unfair coin with p(H) = 0.99, we have

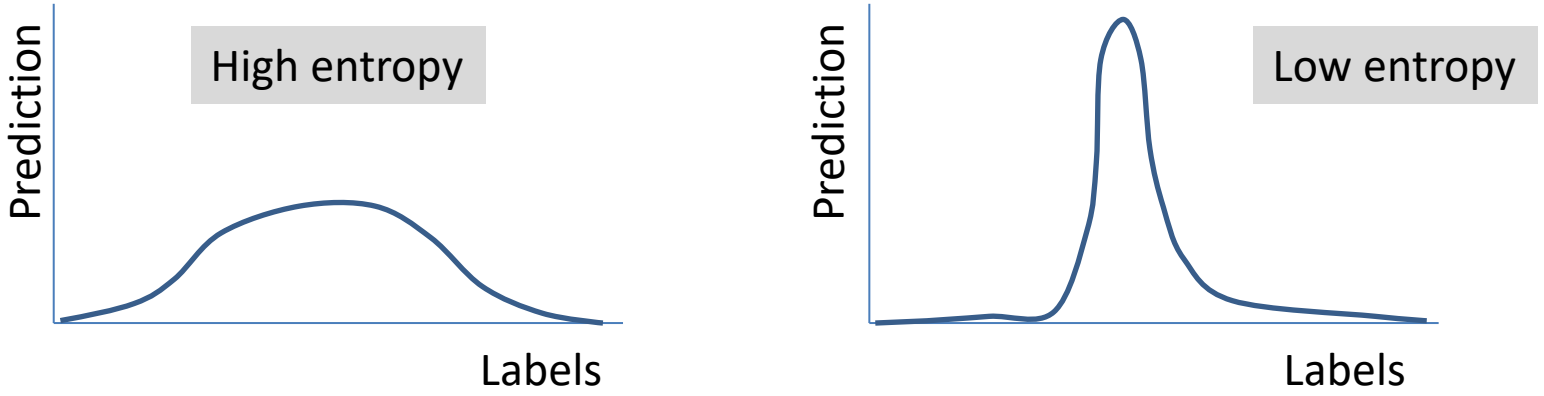
$$H(X) = -(0.99log(0.99) + 0.01log(0.01)) = 0.08bits$$

  - For a fair coin with p(Heads) = 0.5, we have

$$H(X) = -(0.5\ log(0.5) + 0.5\ log(0.5)) = 1\ bit$$

- More uncertainty = Higher entropy
  - The unfair coin delivers very little information (mostly just Heads)

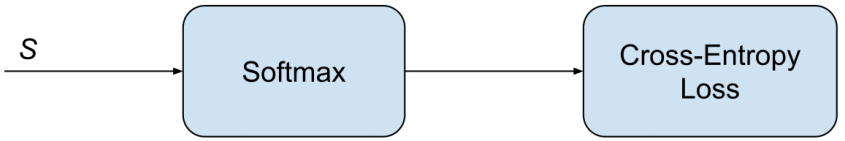
Cross-Entropy and Softmax



- Entropy: optimal way to transmit information about an event
- Cross-entropy: distance between ground truth and predicted distributions

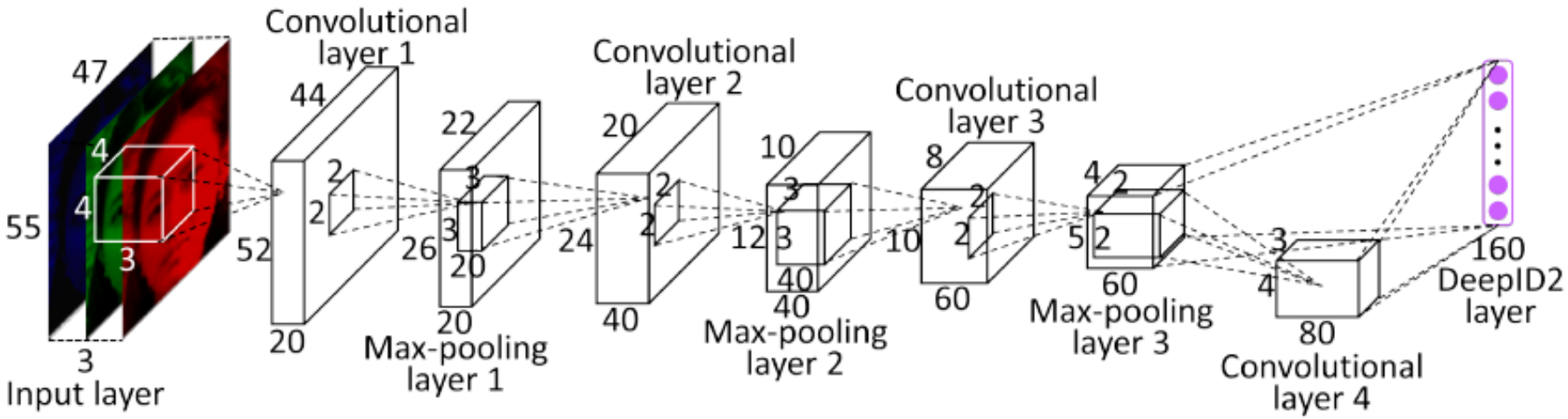
$$H(\bar{D}, D) = -\sum_i P(\bar{D}_i) \log P(D_i)$$

- Softmax loss: cross-entropy on softmax probabilities



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

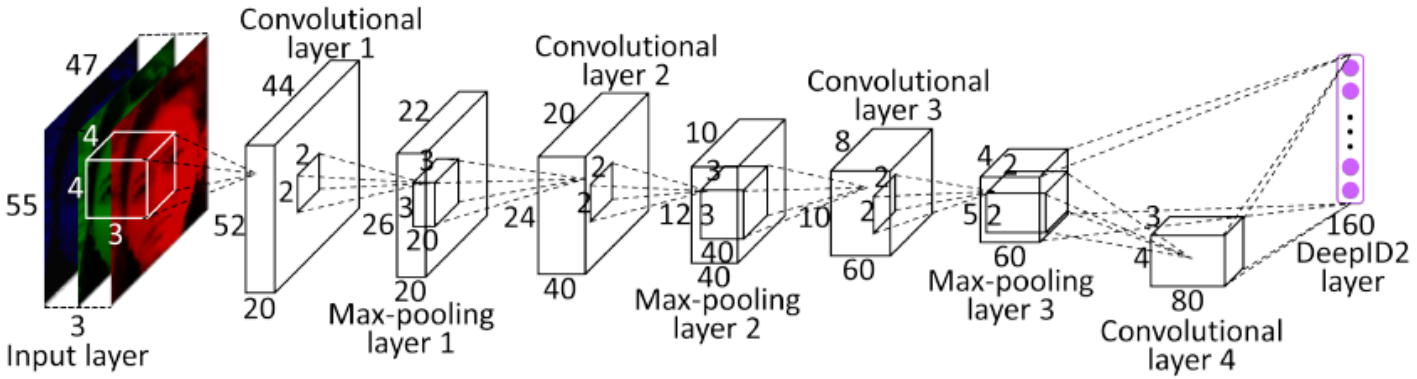
DeepID2: Combine Identification and Verification



- Locally connected layer at the top
  - Respond to facial features at preferred spatial locations
- Feature layer fully connected to last convolutional and locally connected layer
  - Multiscale information
  - Face representation: $f = \mathrm{Conv}(x, \theta_c)$

[Sun et al., DeepID2]

Identification Signal



- Identification: connect feature layer to *n*-way softmax layer
  - Outputs a probability distribution over *n* classes
  - Train with a cross-entropy loss

$$\text{Ident}(f, t, \theta_{id}) = -\sum_{i=1}^{n} -p_i \log \hat{p}_i$$

Feature f

Target class t

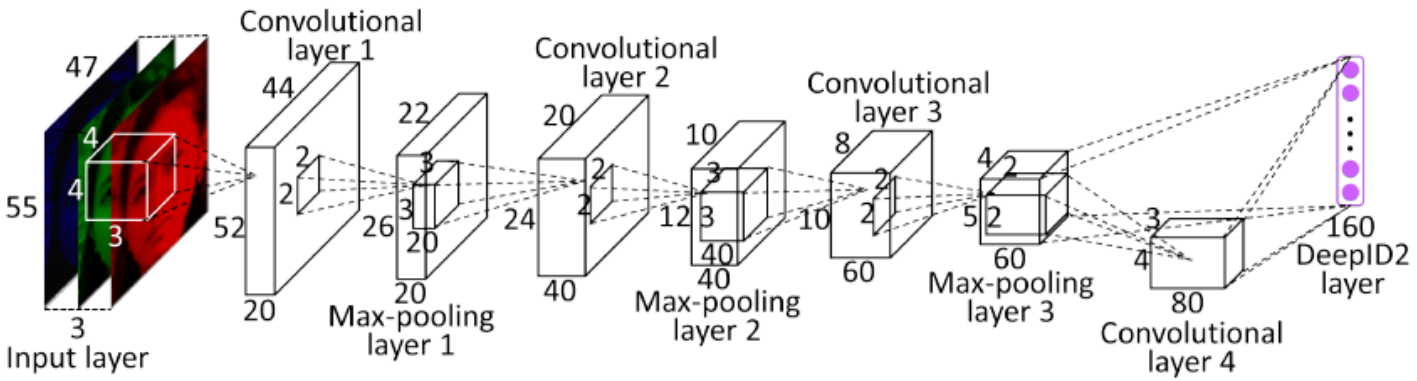
Target probability distribution
p_i = 1 for class t and 0 for other i

Predicted probability distribution

- Goal is to correctly classify all identities simultaneously
  - Incentivize learning discriminative features across inter-personal variations

[Sun et al., DeepID2]

Verification Signal



- Verification: directly regularize the feature vector
  - Pairwise: Gather faces from same class, push those from different classes

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

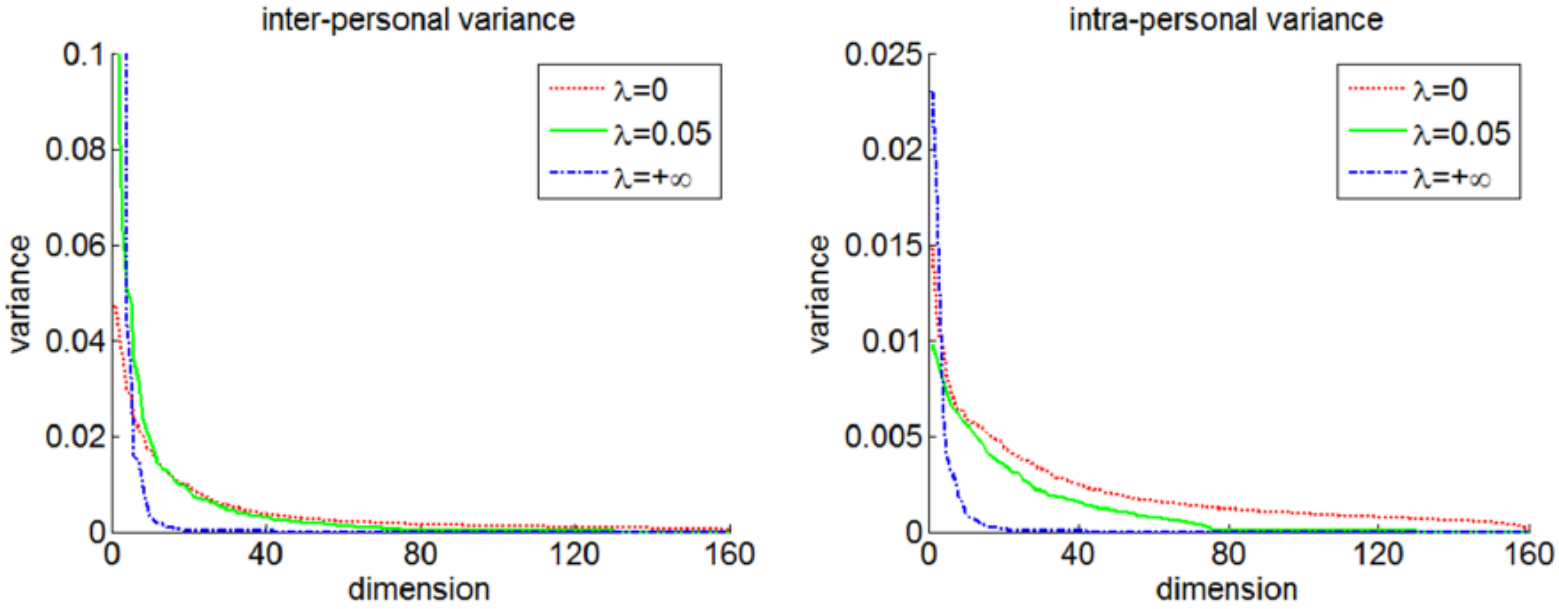
  - Cosine similarity:

$$\text{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \frac{1}{2}\left(y_{ij} - \sigma(wd + b)\right)^2$$, binary y_{ij}, $d = \frac{f_i \cdot f_j}{\|f_i\|_2 \|f_j\|_2}$

- Goal is to learn features that can be matched across intra-personal variations

[Sun et al., DeepID2]

Balancing Identification and Verification



Class *i* mean Dataset mean

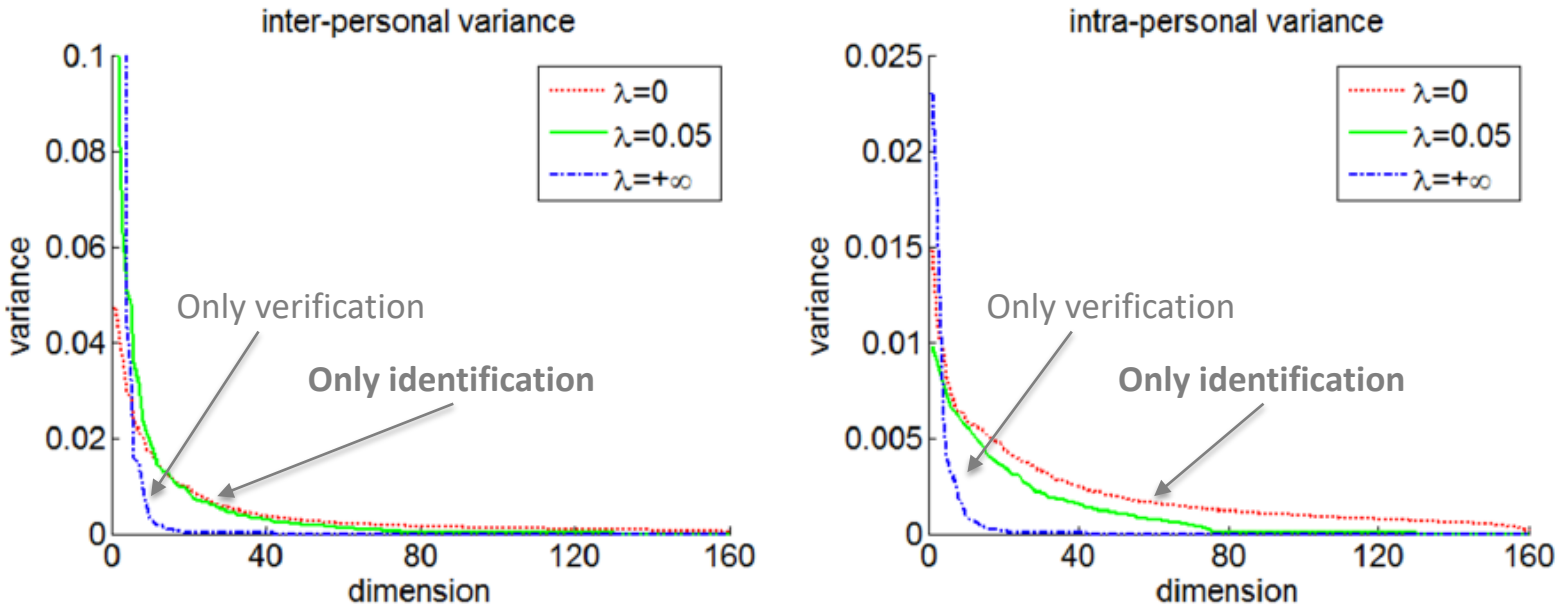
- Inter-class scatter : $\sum_{i=1}^{c} n_i \cdot (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^\top$

- Intra-class scatter : $\sum_{i=1}^{c} \sum_{x \in class\ i} (x - \bar{x}_i)(x - \bar{x}_i)^\top$

c classes,
n_i instances in class *i*

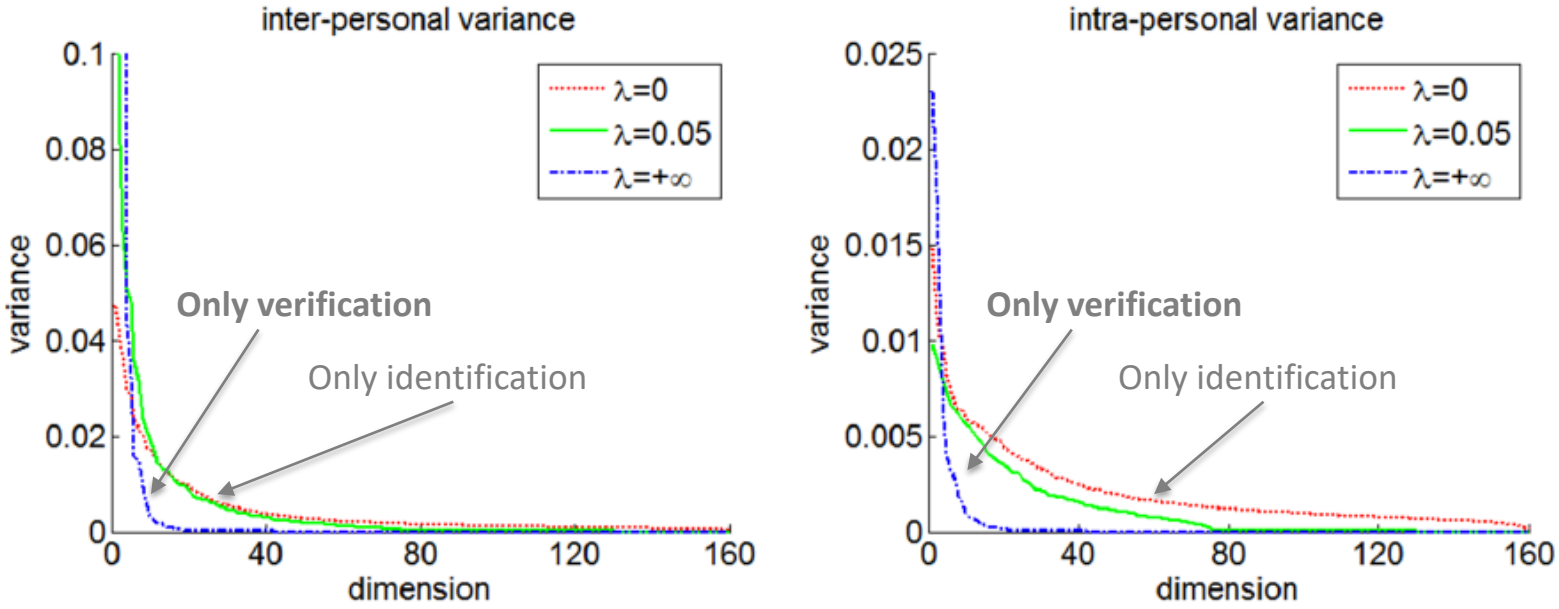
- Variance in scatter indicated by size of eigenvalues
- Small number of eigenvectors: diversity of variation is low
- Both diversity and magnitude of feature variance matters for recognition

Balancing Identification and Verification



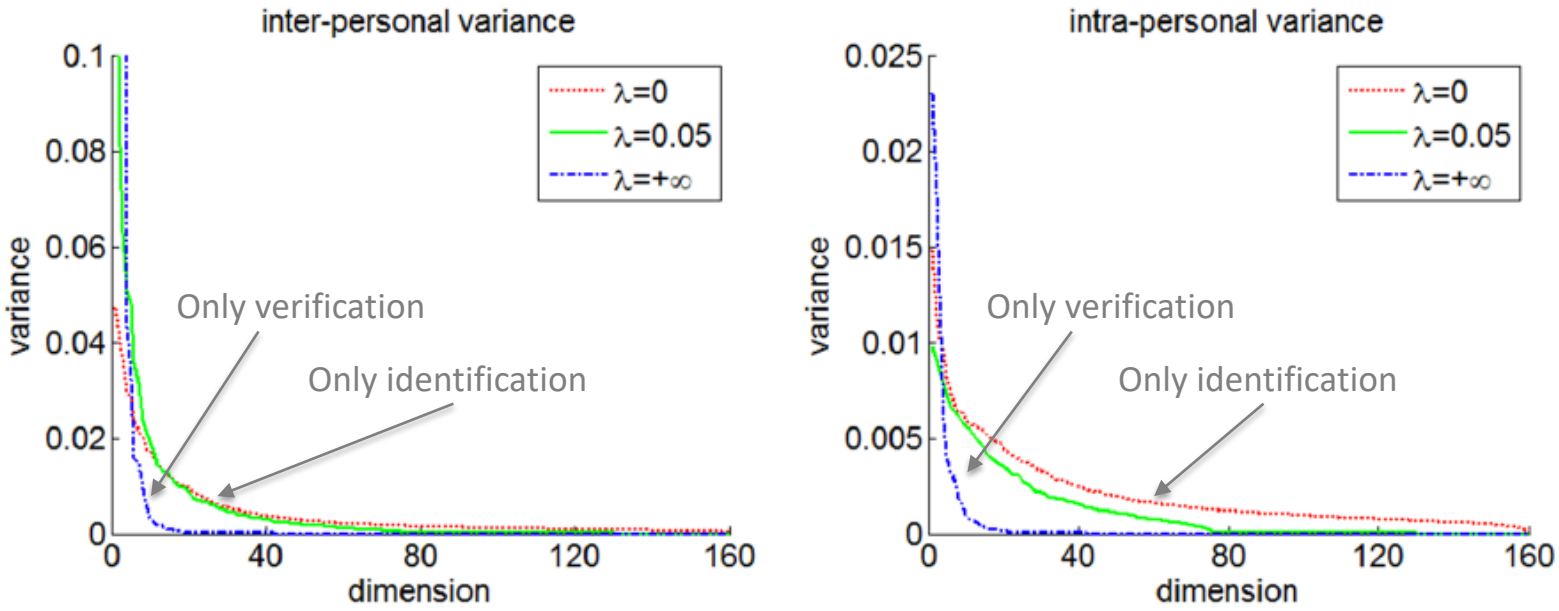
- When only identification signal is used $(\lambda = 0)$:
  - High diversity in both inter-personal and intra-personal features
  - Good for identification since it helps distinguish different identities
  - But large intra-personal variance is noise for verification

Balancing Identification and Verification



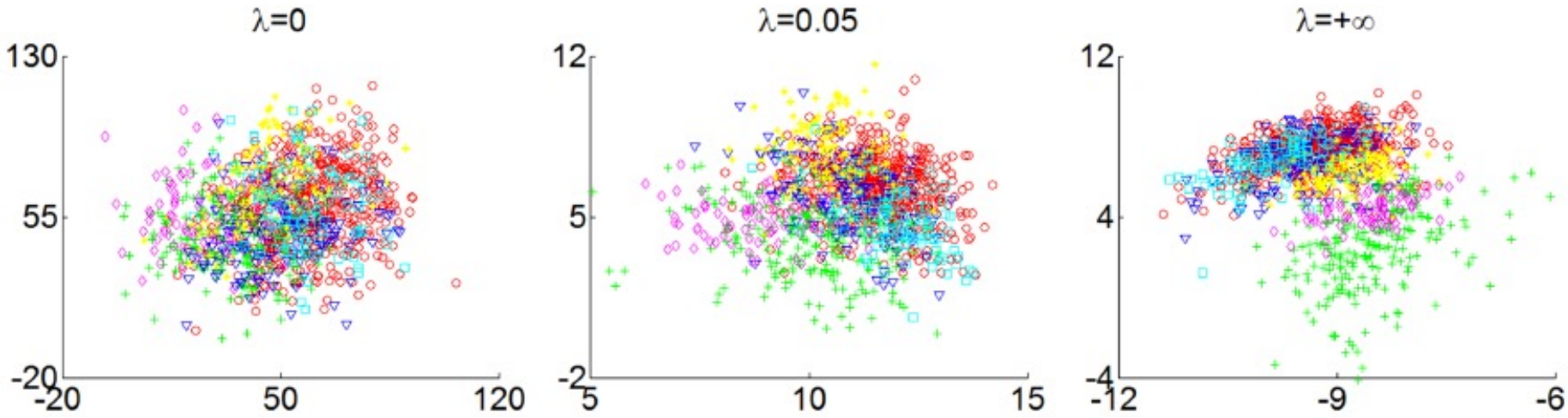
- When only identification signal is used $(\lambda = 0)$:
  - High diversity in both inter-personal and intra-personal features
  - Good for identification since it helps distinguish different identities
  - But large intra-personal variance is noise for verification
- When only verification signal is used (λ approaches $+\infty$):
  - Both intra-personal and inter-personal variance collapse to few directions
  - Cannot distinguish between different identities

Balancing Identification and Verification



- When both verification and identification signals are used $(\lambda = 0.05)$:
  - Inter-personal variations stay high
  - Intra-personal variations reduce in diversity and magnitude

Balancing Identification and Verification



- Visualize features for 6 identities
- With only identification signal:
  - Cluster centers are well-separated, but large cluster size causes overlap
- With only verification signal:
  - Cluster sizes become small, but cluster centers also collapse
- With both signals :
  - Clusters sizes become small and cluster centers are reasonably separated

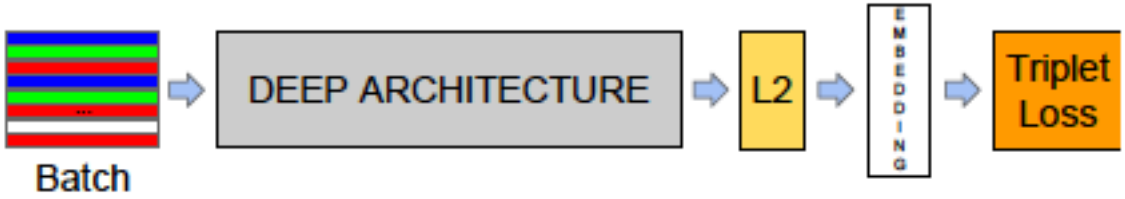
Learn an Embedding for Face Recognition

- **Face verification :** determine whether two images are of the same person
- **Face identification :** determine identity of person in an image
- **Face clustering :** find the same person among a collection of faces

- Train a network such that embedding distances directly represent similarity
  - Faces of same person : small distances
  - Faces of different persons : large distances

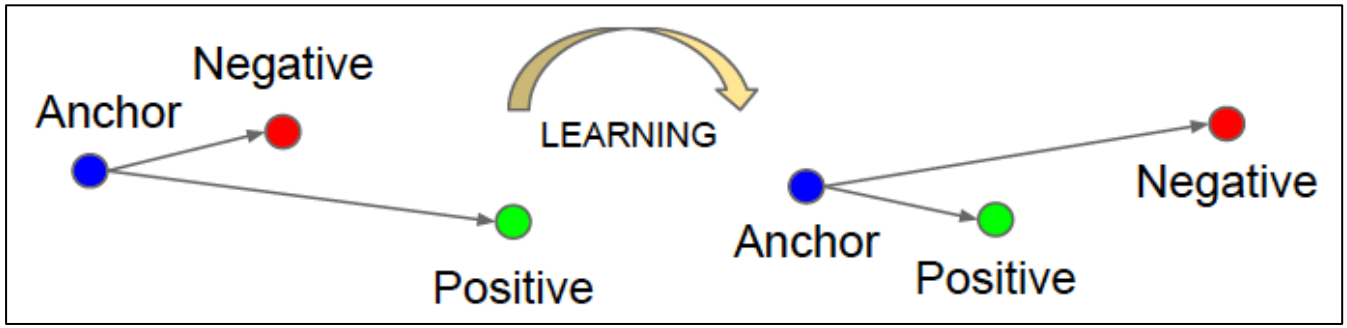
- Once embedding is learned, above problems are all solvable
  - **Verification :** threshold distance between two embeddings
  - **Identification :** can be posed as k-NN classification
  - **Clustering :** can be solved using methods like k-means

FaceNet: Learn an Embedding for Face Recognition



- Goal: learn *d*-dimensional embedding *f(x)* for face image *x*
- Constrain embedding to lie on unit sphere: $\|f(x)\|_2 = 1$

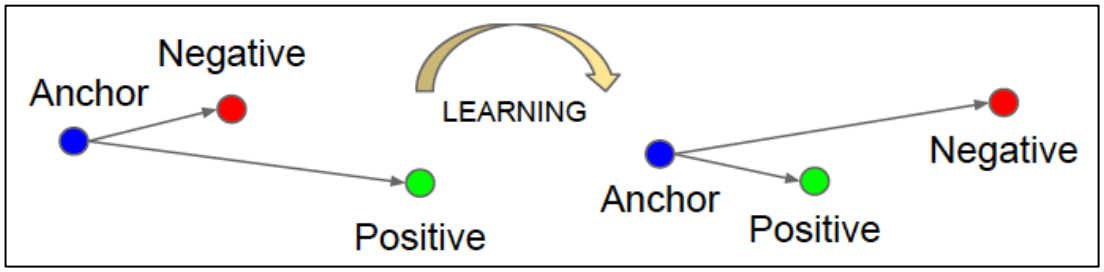
- Goal for triplet loss:
  - Minimize distance between anchor and a positive (from same class)
  - Maximize distance between anchor and a negative (from different classes)



[Schroff et al., FaceNet, CVPR 2015]

Triplet Loss for Training

- Goal for triplet loss:
  - Minimize distance between anchor image x_i^a and a positive x_i^p
  - Maximize distance between anchor x_i^a and a negative x_i^n



$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \quad , \quad \forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in \mathcal{T}$$

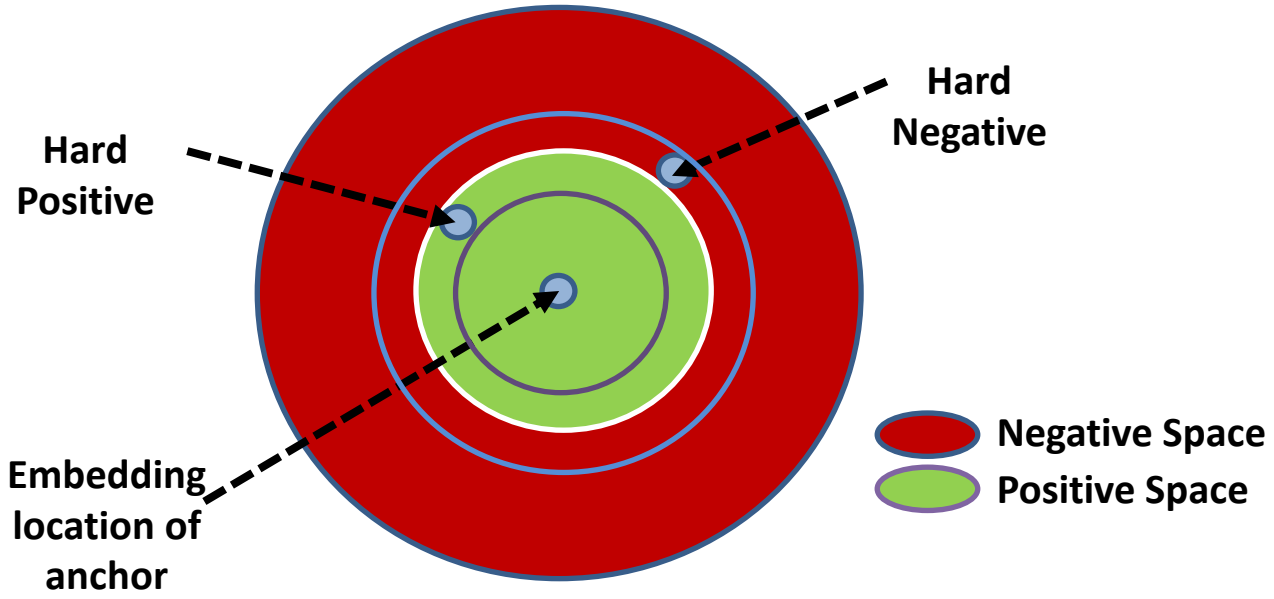
- Total loss to minimize:

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

- Challenge: too many triplets satisfy the margin easily
- Need to select hard examples that are active and improve the model

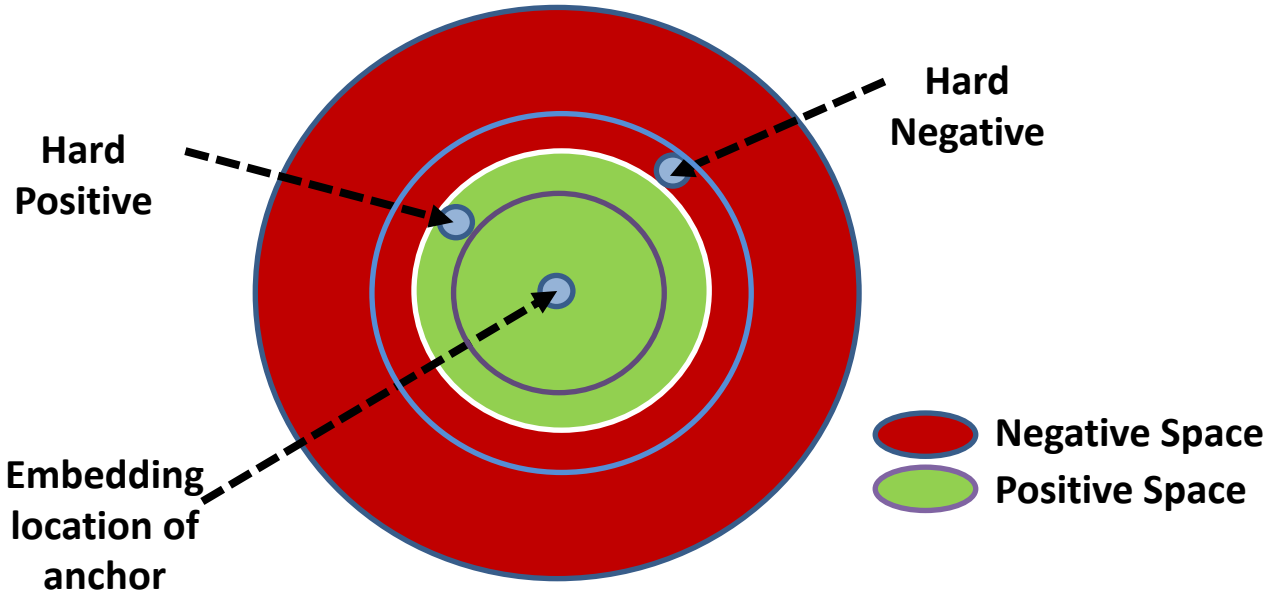
Triplet Selection

- To ensure fast convergence, given an anchor x_i^a :
  - Select **hardest positive** x_i^p such that $\text{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$
  - Select **hardest negative** x_i^n such that $\text{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$



Triplet Selection

- To ensure fast convergence, given an anchor :
  - Select **hardest positive** x_i^p such that $\text{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$
  - Select **hardest negative** x_i^n such that $\text{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$



- **Inefficient** (or infeasible) to compute argmin and argmax over training set
- Might lead to **poor training** as mislabeled or poorly imaged examples dominate

Triplet Selection

- Two courses of action: offline and online selection of triplets
- **Offline:** every n steps, use current feature for argmin and argmax on subset

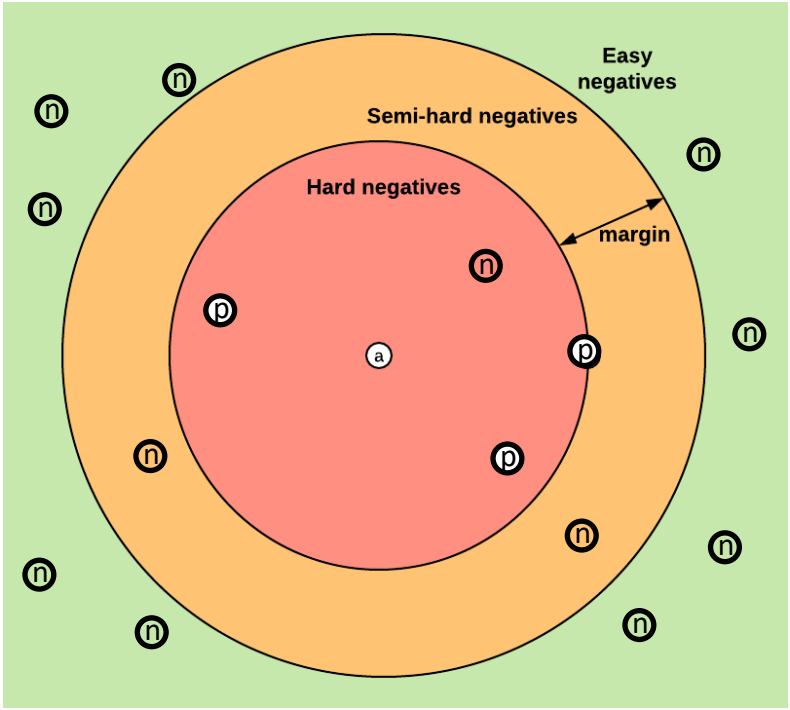
Triplet Selection

- Two courses of action: offline and online selection of triplets

- **Offline:** every n steps, use current feature for argmin and argmax on subset

- **Online:** selecting hard positive and negative examples in mini-batch
  - Use large mini-batch with several thousand examples
  - Use several examples per identity for meaningful anchor-positive distances
  - Randomly sample negatives from other identities

Triplet Selection

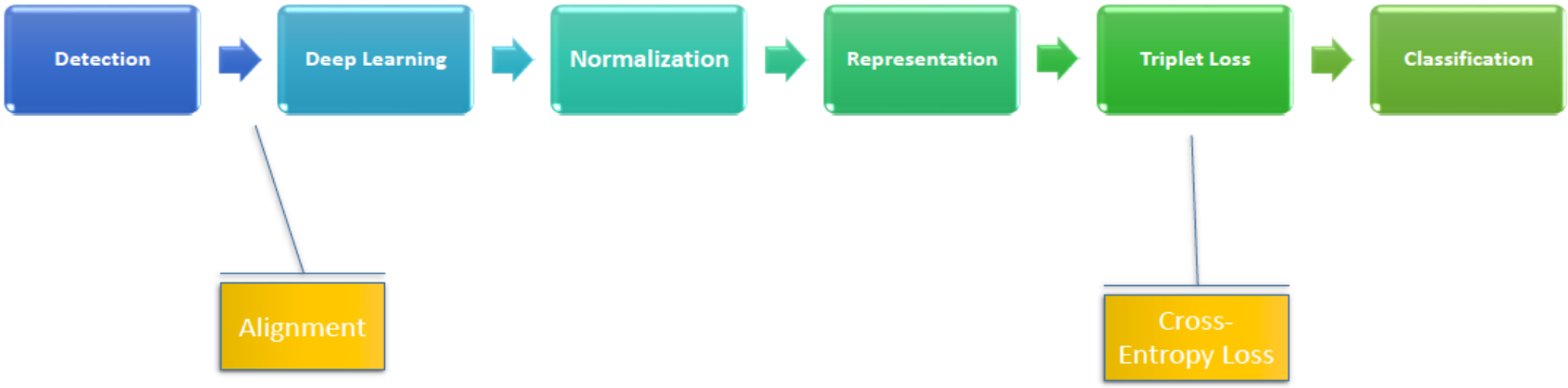


- In practice:
  - Use all anchor-positive pairs, instead of just hard positives
  - Use **semi-hard negatives** at beginning of training

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

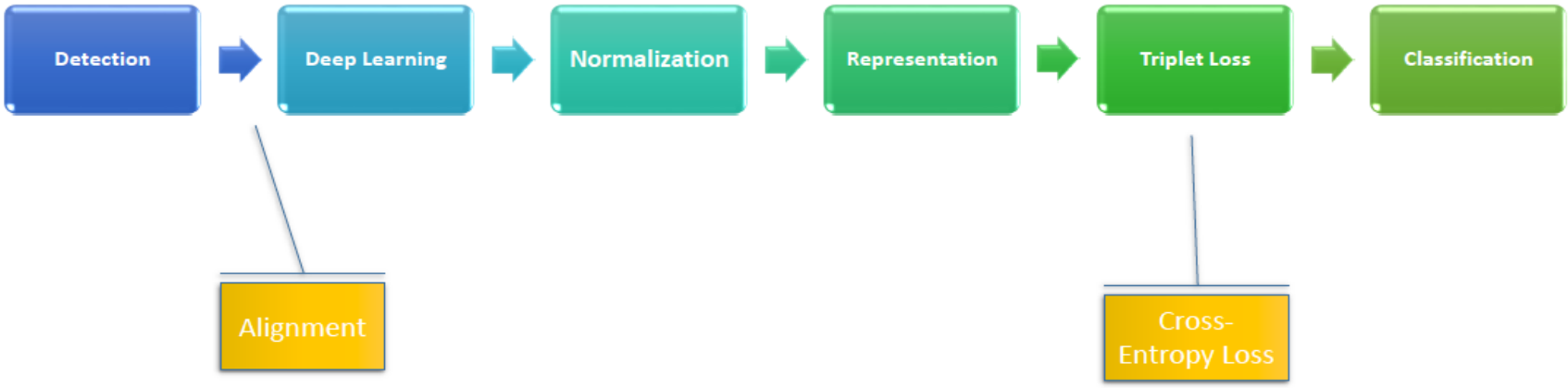
  - Semi-hard negatives: can be within margin, but further than positives
  - Strictly hard negatives at beginning can cause feature to collapse to $f(x) = 0$

Comparison of DeepFace, DeepID2, FaceNet



- Benefit over DeepFace:
  - Learns an embedding, can be used for multiple tasks
  - Only 128-dimensional representation, efficient for inference

Comparison of DeepFace, DeepID2, FaceNet



- Benefit over DeepFace:
  - Learns an embedding, can be used for multiple tasks
  - Only 128-dimensional representation, efficient for inference
- Intuitive benefit of triplet loss over pairwise loss in DeepID2
  - Pairwise: map all faces from one identity to same point
  - Triplet: margin between each pair of faces of an identity and all other faces
  - Triplet loss allows an identity manifold, with distance from other identities