Aprendizado Profundo (Deep Learning)

Similarity Learning

Dario Oliveira (dario.oliveira@fgv.br)

Content

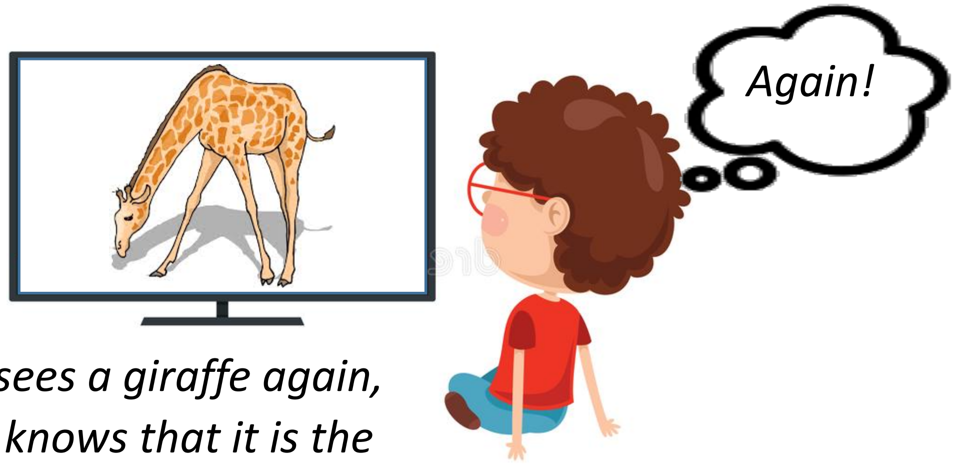
- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks
- Improving Similarity Learning
- Change Detection

Limitation of Classification Approaches

Typically, a classification involves fitting a model given many (some times, hundreds of thousands) examples of each class.

Humans can do better!

Later on

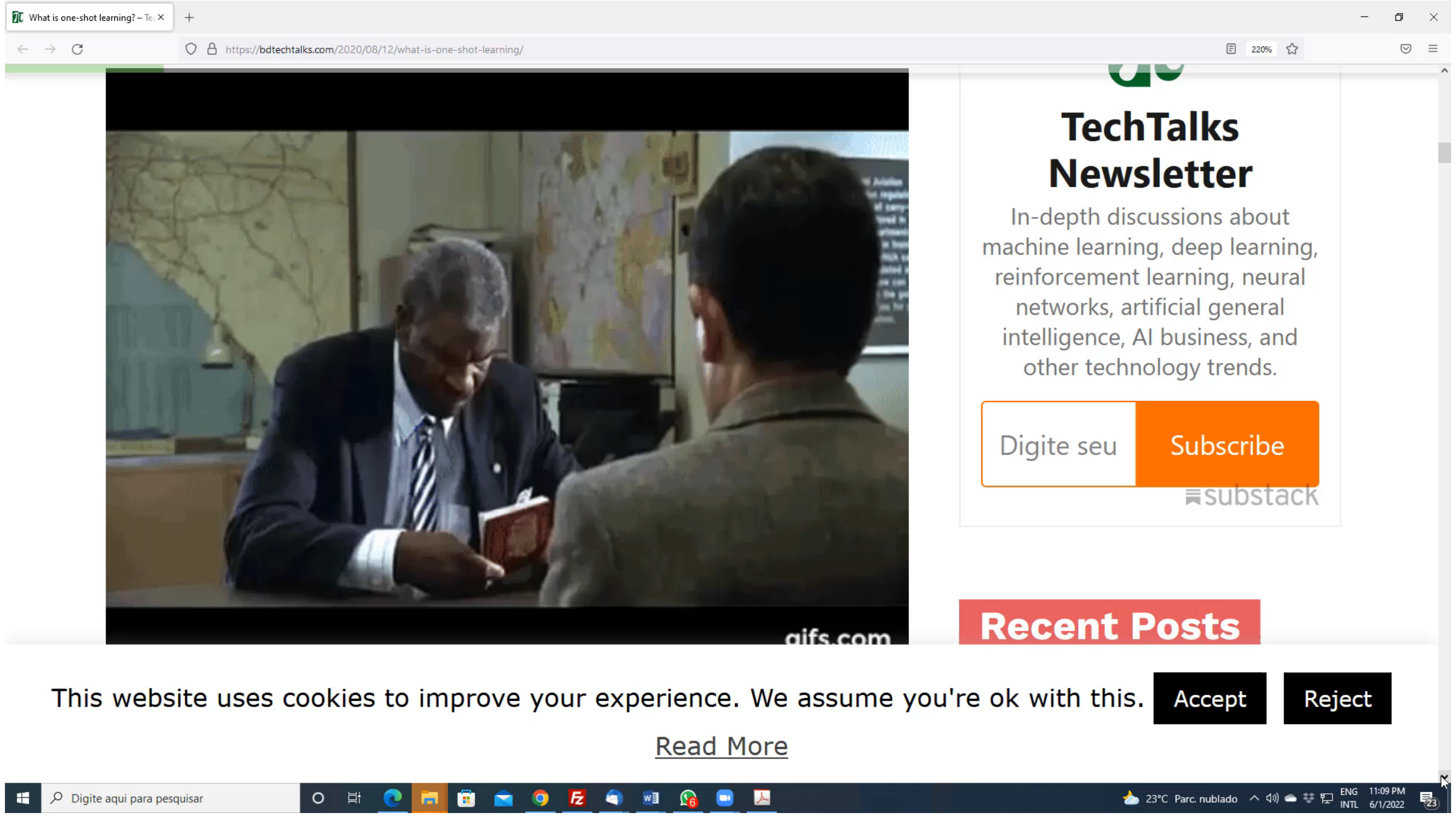


A child knows that it is something unknown when he first sees a giraffe

when he sees a giraffe again, the child knows that it is the same animal he saw earlier.

One-Shot Learning

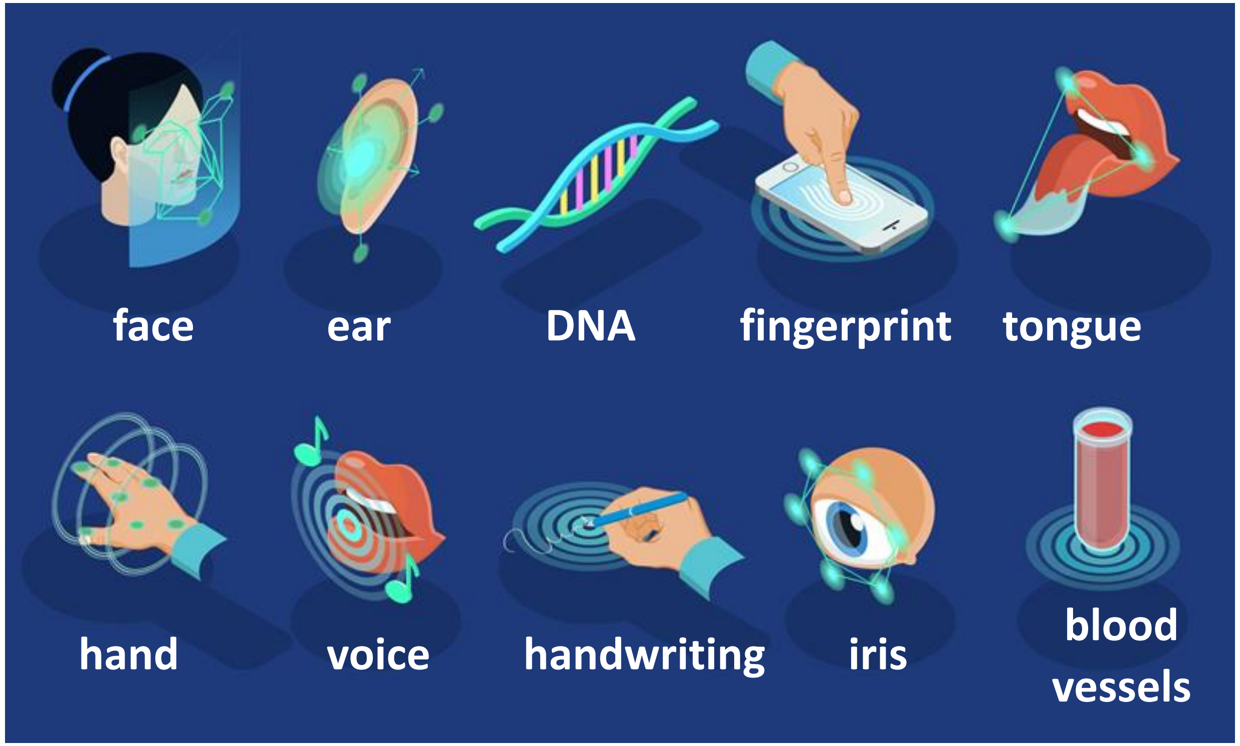
One-shot learning is about looking at two images never seen before and saying whether they represent the same object.



Source: [Ben Dickson, 2020, What is one-shot learning](https://bdtechtalks.com/2020/08/12/what-is-one-shot-learning/)

One-Shot Learning applications

Biometrics



Source: [HR]

One-Shot Learning applications

Recommendation systems

Source: [Anchit Jain]

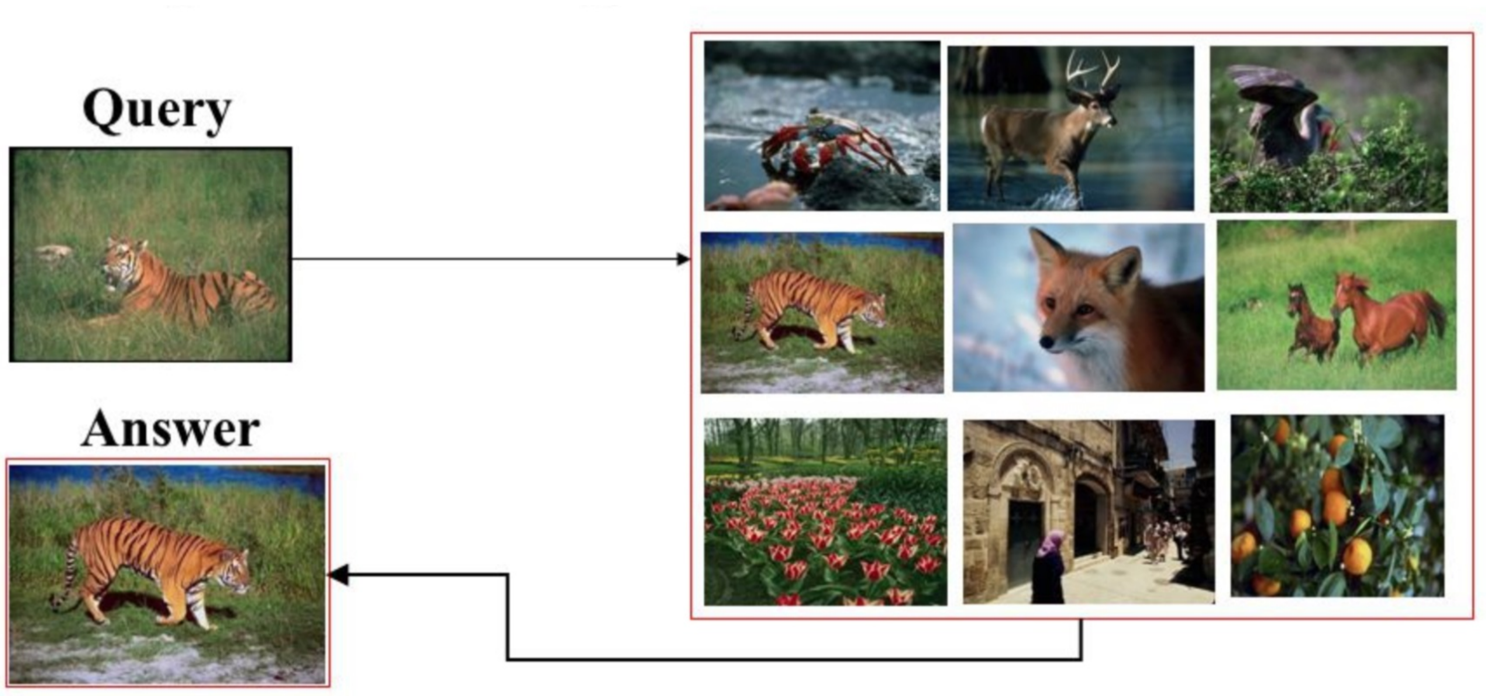
One-Shot Learning applications

Matching résumés to job descriptions

Source: LADDERS

One-Shot Learning applications

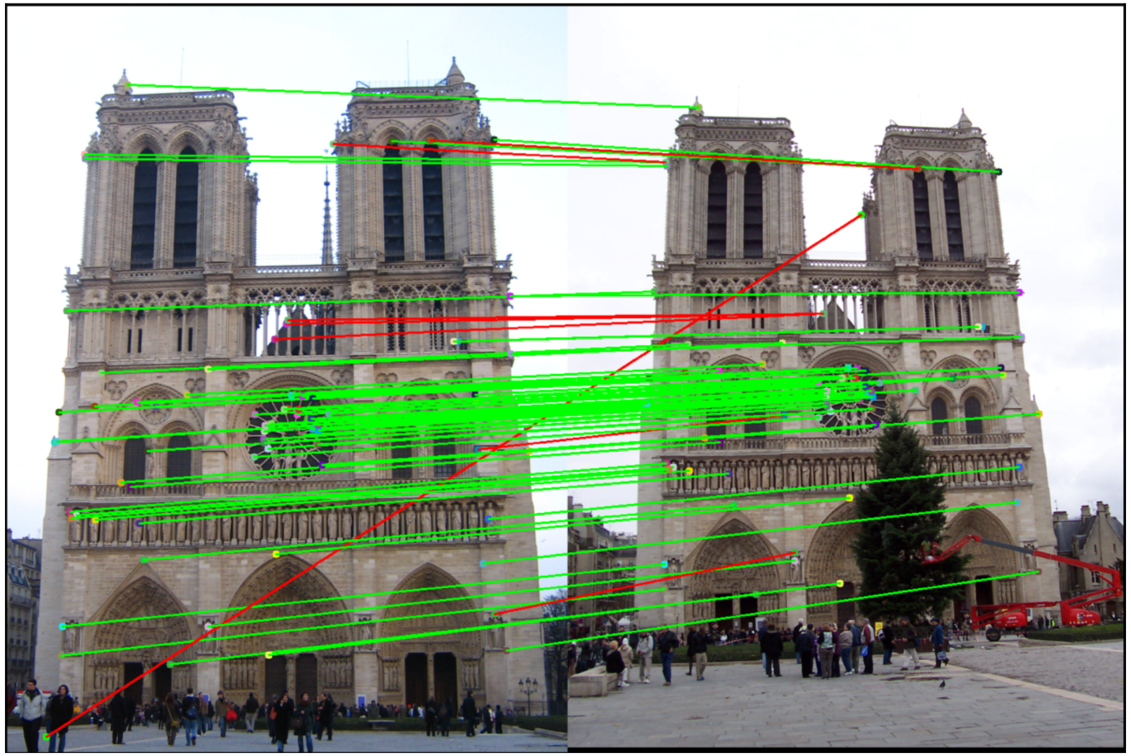
Content-based image retrieval



Source: [Gurchetan Singh]

One-Shot Learning applications

Image Matching



Source: https://www.cc.gatech.edu/~hays/compvision/proj2/

Content

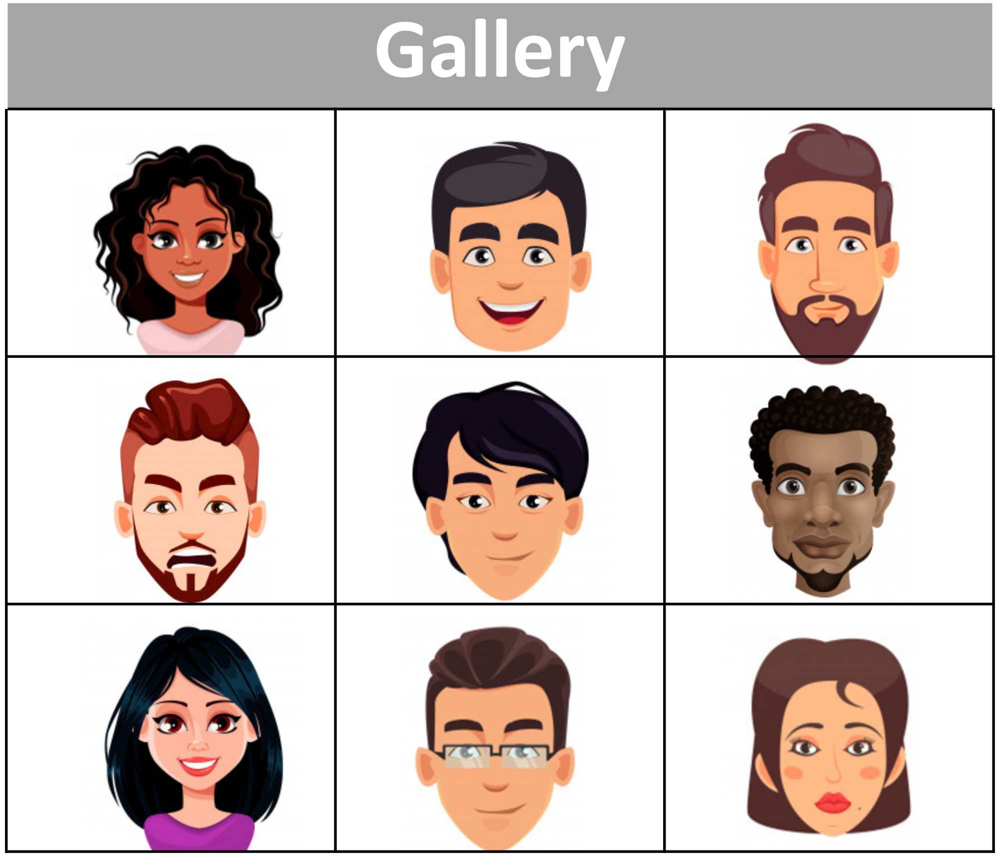
- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks
- Improving Similarity Learning
- Change Detection

Face Recognition

Identification problem: who is the subject?

A possible solution:

- ➢Define a class for each subject in the gallery.
- ➢Train a classifier
- ➢Use the trained classifier to answer the question.



What if you add a new subject in the gallery?

Repeat all steps again!

Face Recognition

Verification problem: has the subject the alleged identity?

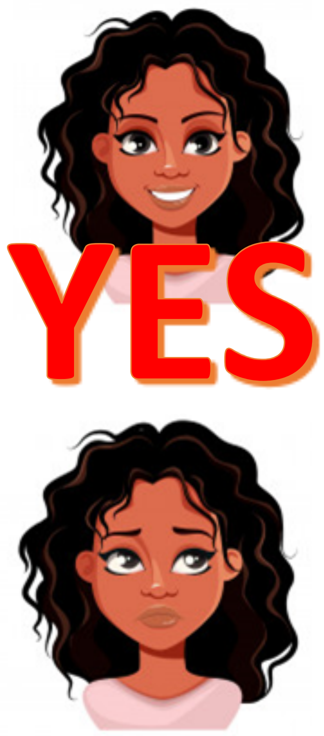
Are they the same person?

A

B

C

D



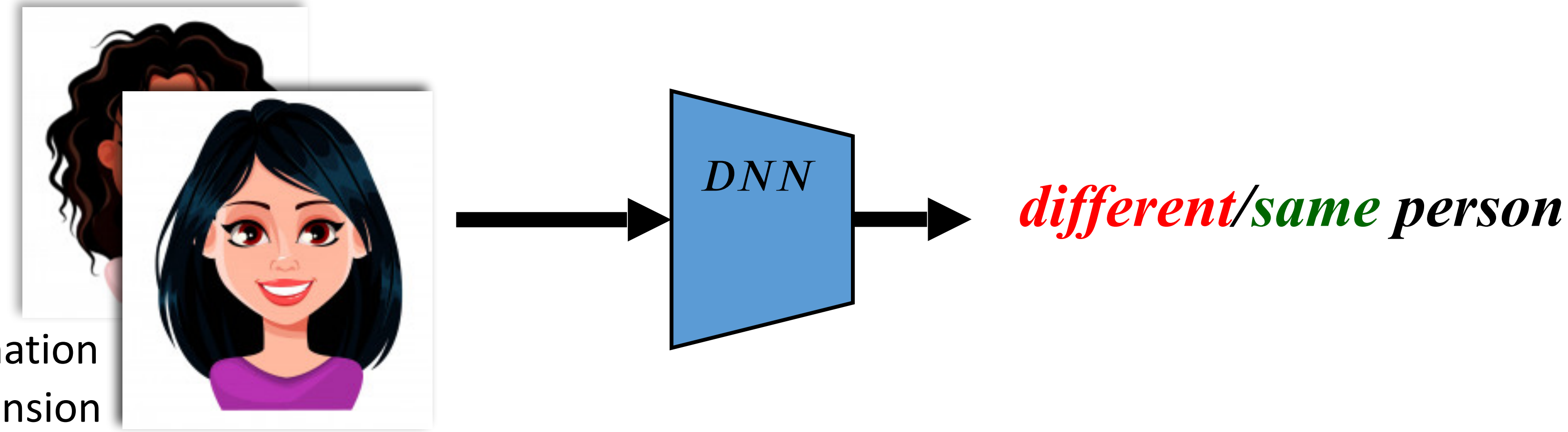
Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks
- Improving Similarity Learning
- Change Detection

Early Fusion

A simple solution for verification: transform into a binary classification problem

typically, concatenation along the 3rd dimension



use the binary cross-entropy as loss function.

Similarity Learning

Instead of learning how to classify the input – **learn how to measure similarity** of two inputs

Are they the same person?

Low similarity score

High similarity score

Similarity Learning

Learning Distance Metric

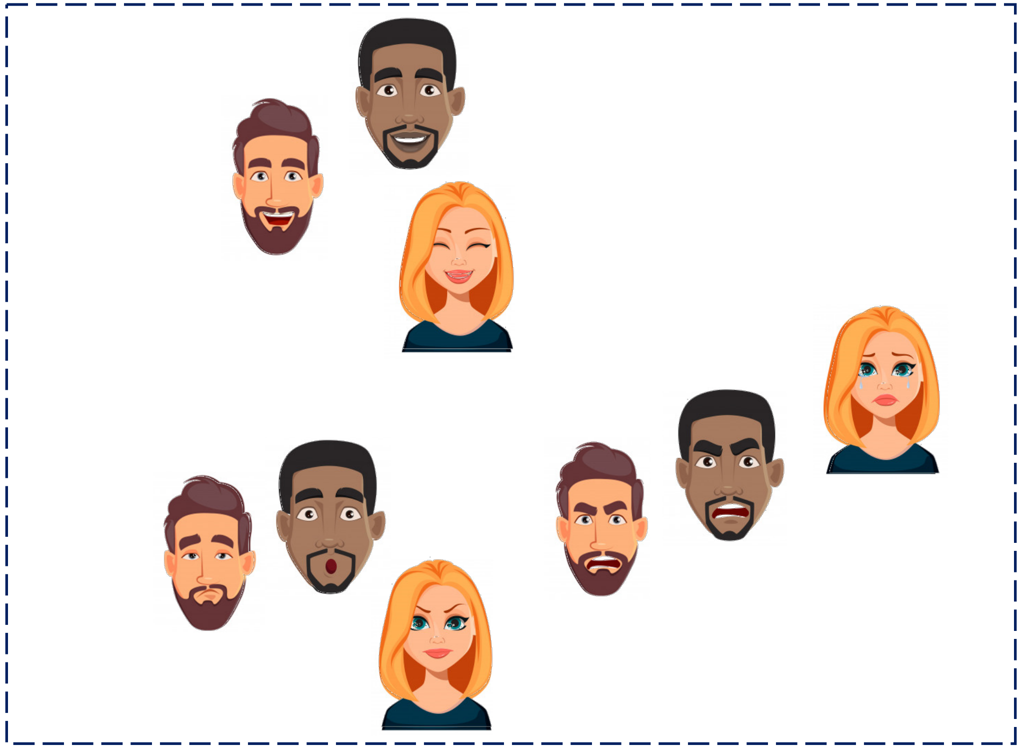
- An active research topic in CV community.
- Learn a **distance function** that assigns small/large distance to pairs of examples that are semantically similar/dissimilar.

Learning Feature Embeddings

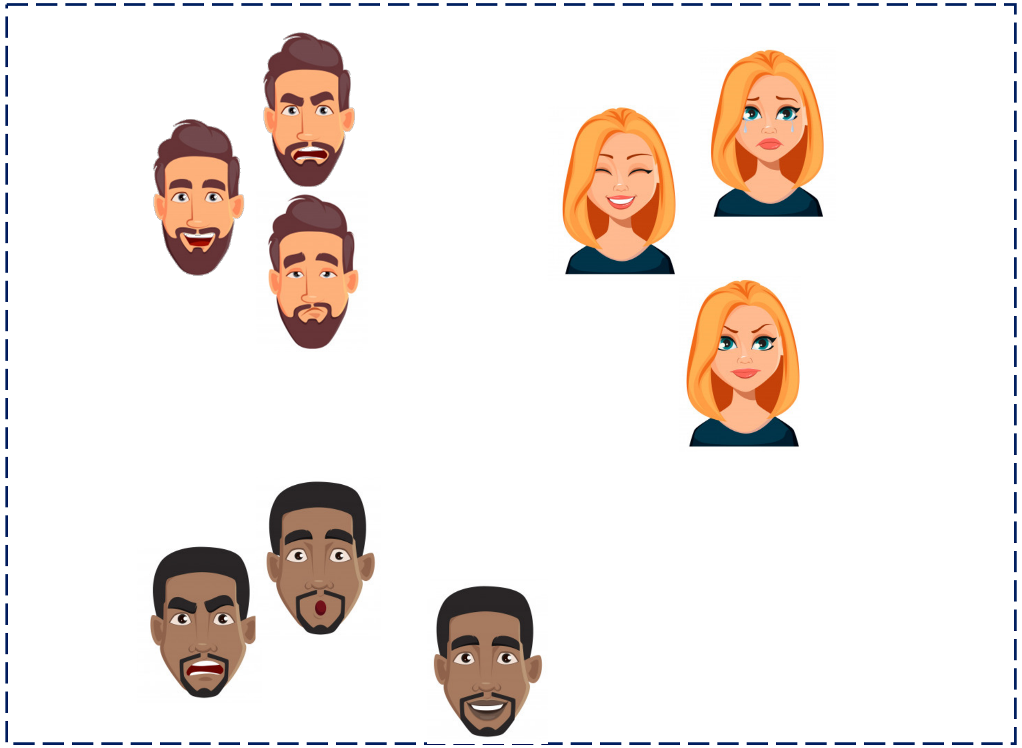
- With the advent of DNNs, metric learning turned to learning **feature embeddings** that better fits a simple distance function, such as Euclidean distance or cosine distance.

Similarity Learning

Is about learning an embedding space $\mathbb{R}^k$ where images of the same person are close and of different persons are far away.



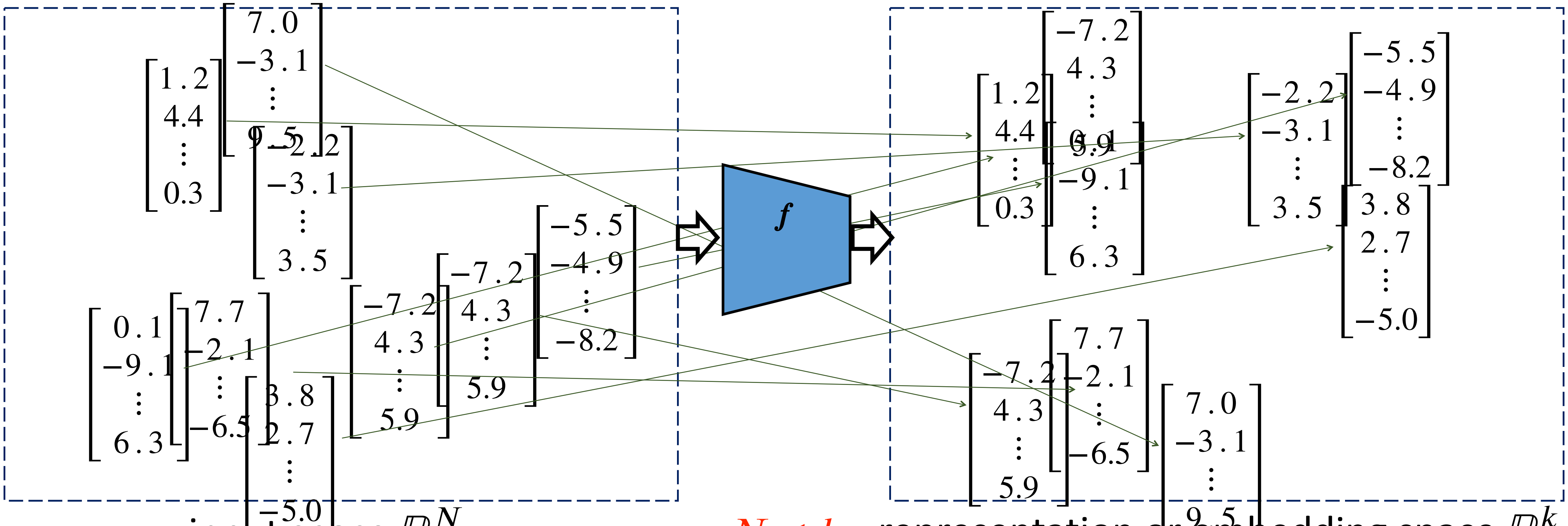
input space $\mathbb{R}^N$



representation or embedding space $\mathbb{R}^k$

Learning an Embedding Space

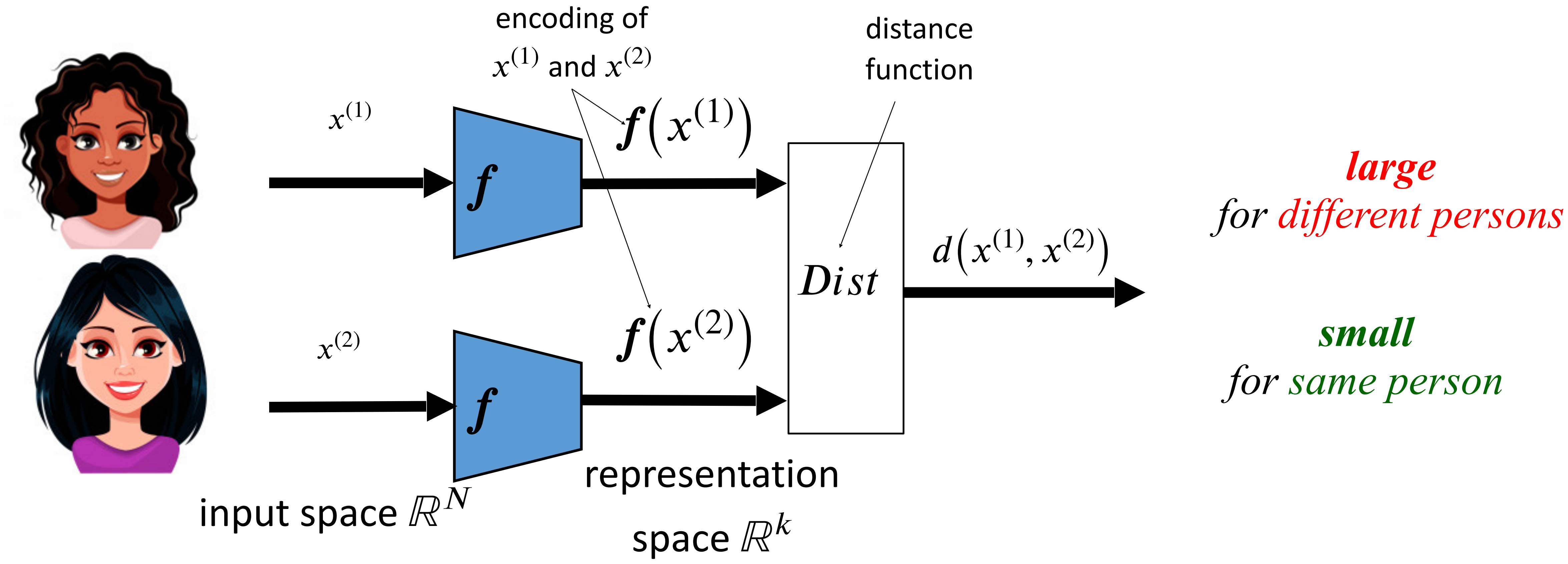
Step 1: find a mapping $\boldsymbol{f}: \mathbb{R}^N \rightarrow \mathbb{R}^k$ to an embedding space where images of the same person are close and of different persons are far away.



input space $\mathbb{R}^N$ $\quad N < k \quad$ representation or embedding space $\mathbb{R}^k$

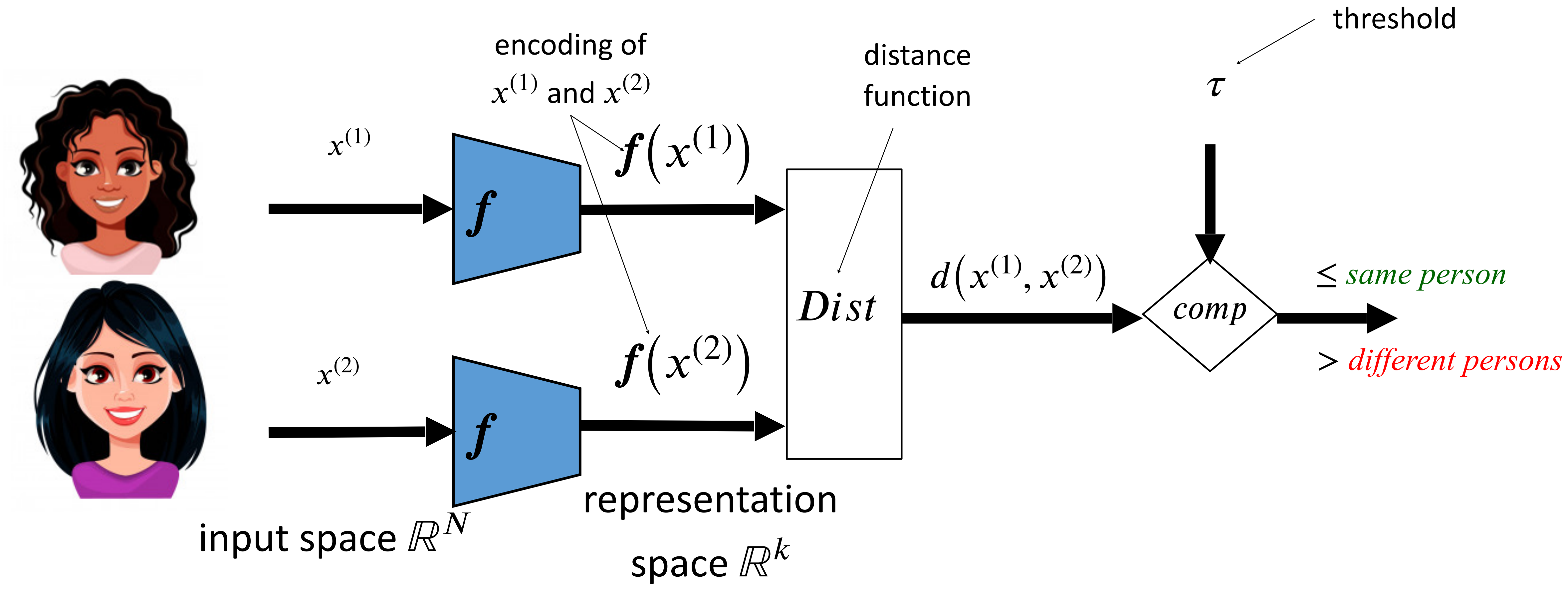
Distance Learning

Step 2: choose a distance function $Dist$ in the embedding space related to the (dis)similarity between input images $x^{(1)}$ and $x^{(2)}$.



Choose a Threshold

Step 3: define a threshold τ to decide whether the inputs are of the same or of different persons.



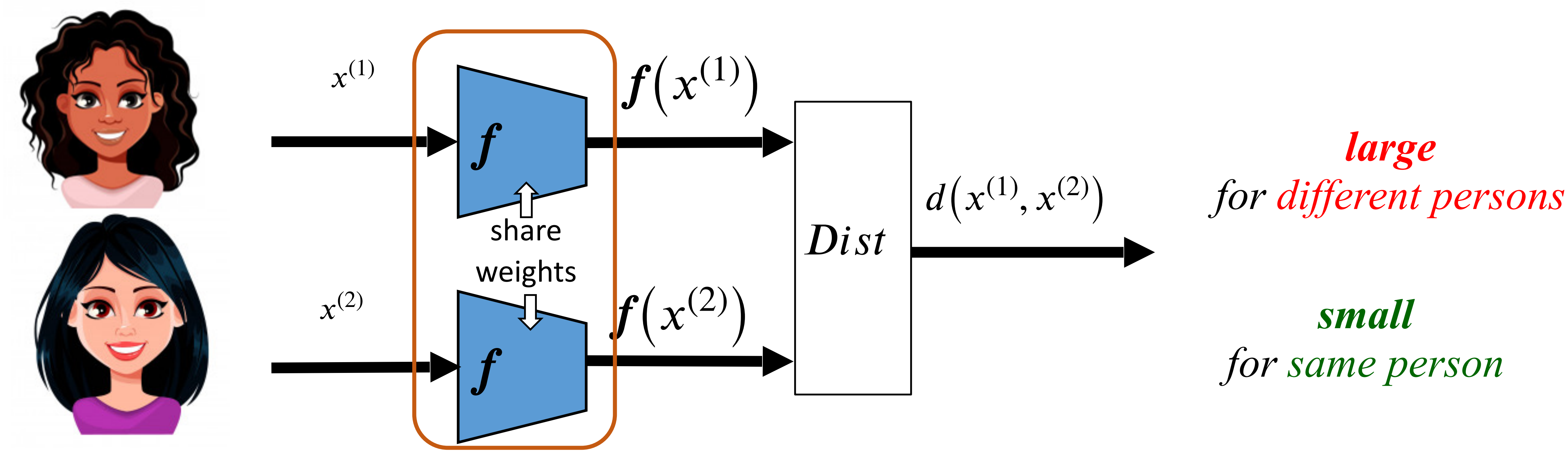
Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks
- Improving Similarity Learning
- Change Detection

Siamese Network

- Two identical networks that share their weights.
- Distance function (e.g., Euclidean)

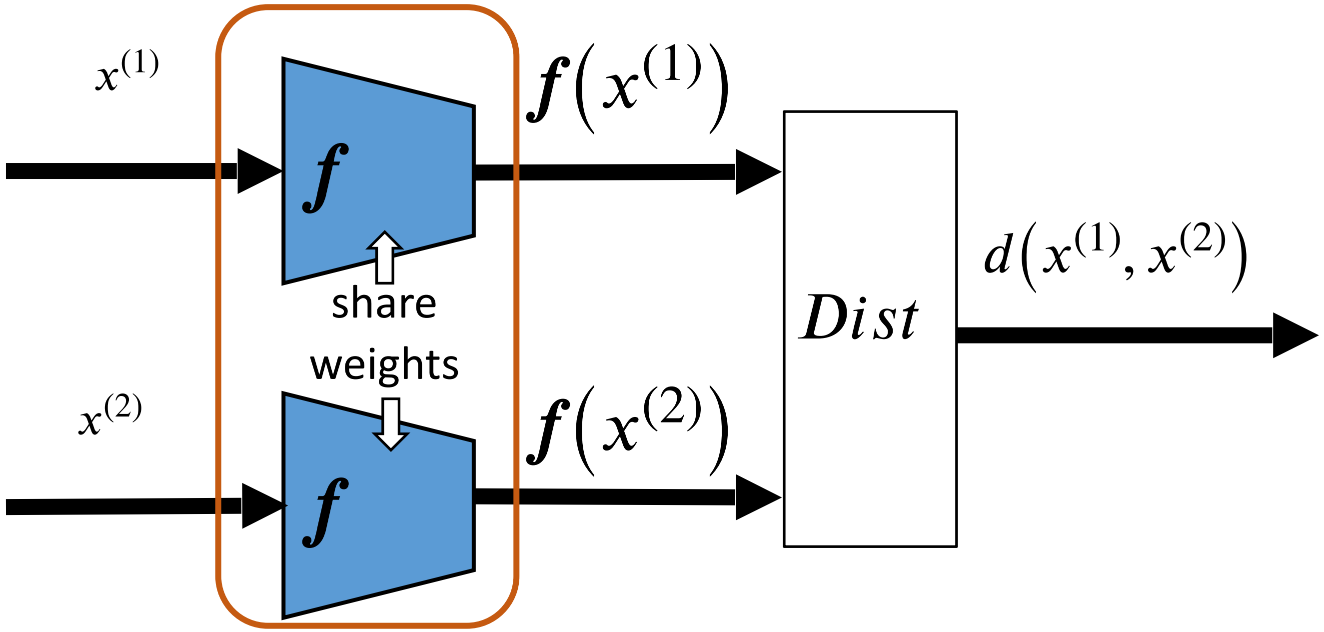
$$d\left(x^{(1)}, x^{(2)}\right) = \left\| \boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right) \right\|^2$$



Taignan et al., 2014, DeepFace: closing the gap to human level performance

Training a Siamese network

Update the weights for each branch independently, and average them.



Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks – Contrastive Loss
- Improving Similarity Learning
- Change Detection

Contrastive Loss

For a **similar (positive)** pair

- If A and B are from the same person, $d(A, B)$ should be small

$x^{(1)}$

$x^{(2)}$

$$\mathscr{L}\left(x^{(1)}, x^{(2)}\right) = \left\| \boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right) \right\|^2$$

Contrastive Loss

For a **dissimilar (negative)** pair

- If A and B are from different persons, $d(A, B)$ should be large
- and set the loss to zero if $d(A, B)$ exceeds a margin m^2

$x^{(1)}$

$x^{(2)}$

$$\left\| f\left(x^{(1)}\right) - f\left(x^{(2)}\right) \right\|^2 > m^2, \text{ for some } m$$

margin

$$m^2 < \left\| f\left(x^{(1)}\right) - f\left(x^{(2)}\right) \right\|^2$$

$$m^2 - \left\| f\left(x^{(1)}\right) - f\left(x^{(2)}\right) \right\|^2 < 0$$

$$\mathcal{L}\left(x^{(1)}, x^{(2)}\right) = max\left(0, m^2 - \left\| f\left(x^{(1)}\right) - f\left(x^{(2)}\right) \right\|^2\right)$$

aka Hinge loos

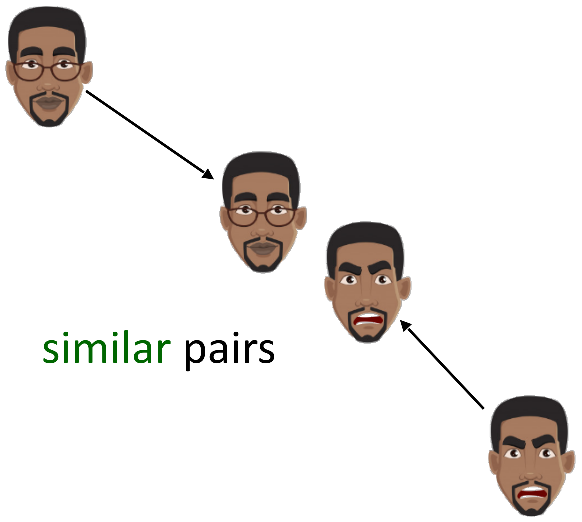
Contrastive Loss

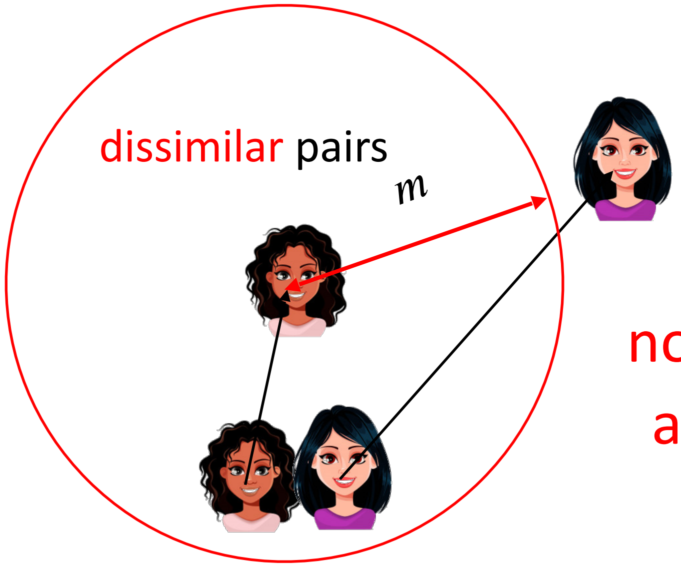
The general formulation for **any single** pair $\left(x^{(1)}, x^{(2)}\right)$:

$$\mathcal{L} = y\left\|\boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right)\right\|^2 + (1-y)\,max\left(0, m^2 - \left\|\boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right)\right\|^2\right)$$

$y = 0$ for dissimilar pairs

$y = 1$ for similar pairs





no effort to move away beyond m

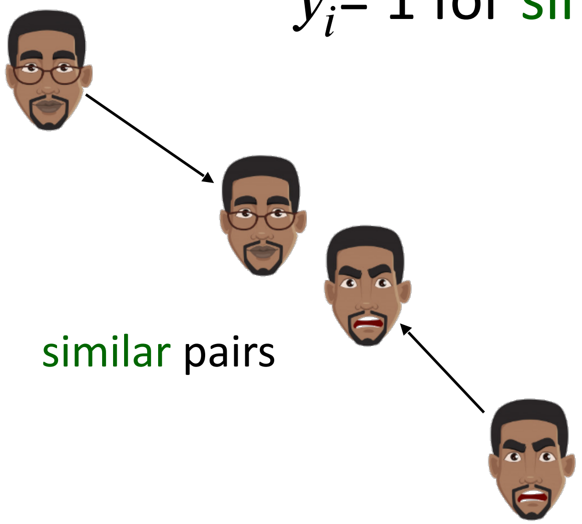
Contrastive Loss

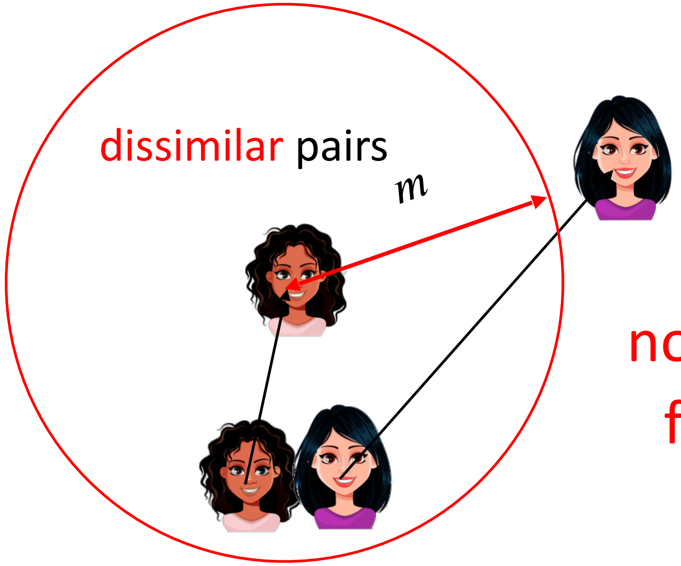
Formulation considering **all** pairs $\left\{ \left(x_i^1, x_i^2\right) \right\}$ in a batch :

$$\boxed{\mathscr{L} = \sum_i y_i \left\| \boldsymbol{f}\left(x_i^{(1)}\right) - \boldsymbol{f}\left(x_i^{(2)}\right) \right\|^2 + \left(1 - y_i\right) max\left(0, m^2 - \left\| \boldsymbol{f}\left(x_i^{(1)}\right) - \boldsymbol{f}\left(x_i^{(2)}\right) \right\|^2\right)}$$

y_i= 0 for dissimilar pairs

y_i= 1 for similar pairs





no effort to move further than m

Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks - WDMC
- Improving Similarity Learning
- Change Detection

Weighted Double Margin Contrastive Loss

For a **similar (positive)** pair

- If A and B are from the same person, $d(A, B)$ should be small
- and set the loss to zero if $d(A, B)$ is lower than a margin m_1^2

$x^{(1)}$

$$\mathcal{L}\left(x^{(1)}, x^{(2)}\right) = max\left(0, \left\|\boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right)\right\|^2 - m_1^2\right)$$

$x^{(2)}$

Chen et al., 2020, DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images

Weighted Double Margin Contrastive Loss

For a **dissimilar (negative)** pair

- If A and B are from different persons, $d(A, B)$ should be large
- and set the loss to zero if $d(A, B)$ <u>exceeds a</u> margin m_2^2

exactly the same as for the contrastive loss

$x^{(1)}$

$x^{(2)}$

$$\mathscr{L}\left(x^{(1)}, x^{(2)}\right) = max\left(0, m_2^2 - \left\| \boldsymbol{f}\left(x^{(1)}\right) - \boldsymbol{f}\left(x^{(2)}\right) \right\|^2\right)$$

Chen et al., 2020, DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images

Weighted Double Margin Contrastive Loss

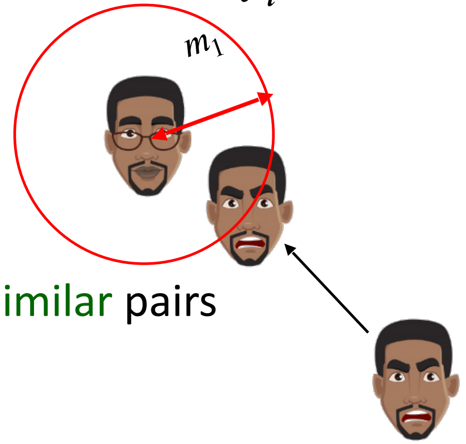
Formulation considering **all** pairs $\left\{\left(x_i^{(1)}, x_i^{(2)}\right)\right\}$ in a batch :

w_1, w_2 are the weights for
similar, dissimilar pairs

$$\mathcal{L} = \sum_i w_1 y_i max\left(0, \left\|f\left(x_i^{(1)}\right) - f\left(x_i^{(2)}\right)\right\|^2 - m_1^2\right) + w_2(1 - y_i)max\left(0, m_2^2 - \left\|f\left(x_i^{(1)}\right) - f\left(x_i^{(2)}\right)\right\|^2\right)$$

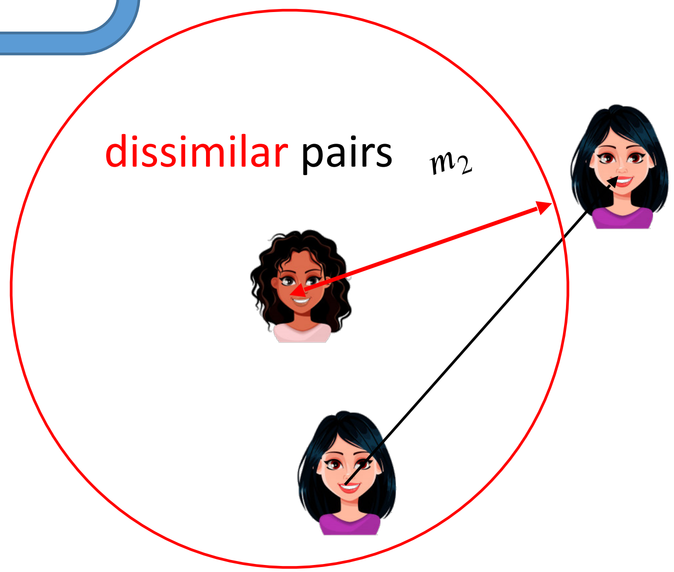
$y_i = 0$ for dissimilar pairs
$y_i = 1$ for similar pairs

no effort to move
closer than m_1



similar pairs

dissimilar pairs



no effort to move
further than m_2

Chen et al., 2020, DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images

Weighted Double Margin Contrastive loss

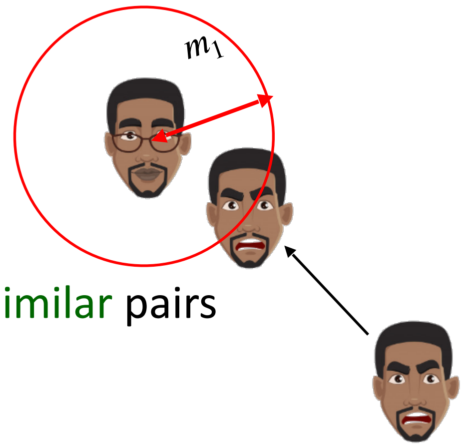
Formulation considering **all** pairs $\left\{ \left(x_i^{(1)}, x_i^{(2)} \right) \right\}$ in a batch :

$$\boxed{\mathscr{L} = \sum_i y_i w_1 max\left(0, \left\| \boldsymbol{f}\left(x_i^{(1)}\right) - \boldsymbol{f}\left(x_i^{(2)}\right) \right\|^2 - m_1^2\right) + (1 - y_i) w_2 max\left(0, m_2^2 - \left\| \boldsymbol{f}\left(x_i^{(1)}\right) - \boldsymbol{f}\left(x_i^{(2)}\right) \right\|^2\right)}$$

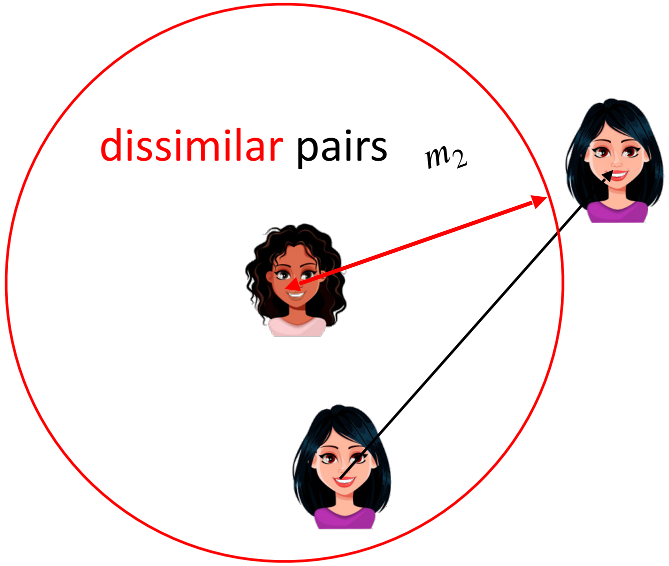
y_i= 0 for dissimilar pairs

y_i= 1 for similar pairs

no effort to move closer than m_1



similar pairs



no effort to move further than m_2

[Chen et al., 2020, DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images]()

Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks – Triplet Loss
- Improving Similarity Learning
- Change Detection

Triplet loss

Training is conducted on triplets instead of pairs:

Anchor (A)
any arbitrary sample

Positive (P)
same class as the anchor

Negative (N)
different class from the anchor

We want $\left\| f(A) - f(P) \right\|^2 < \left\| f(A) - f(N) \right\|^2$

Schorff et al., 2015 - FaceNet: A Unified Embedding for Face Recognition and Clustering

Triplet loss

For a single triplet:

$$\|f(A) - f(P)\|^2 < \|f(A) - f(N)\|^2$$

$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 < 0$$

margin

Making it more restrictive

$$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 < -m$$

for some $m > 0$

$$\boxed{\mathscr{L}(A, P, N) = max\left(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + m\right)}$$

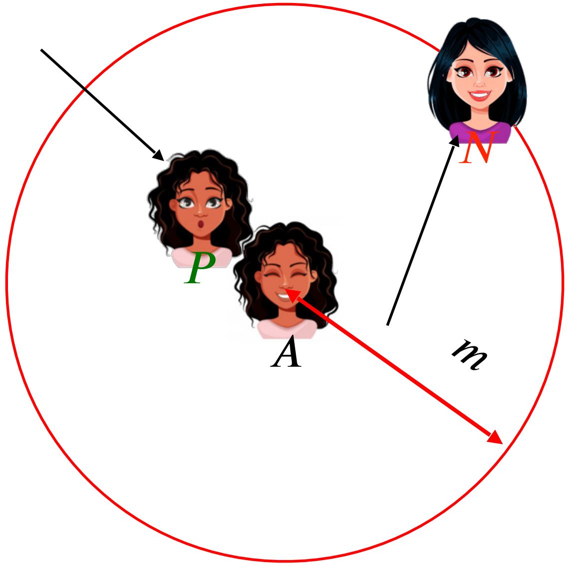
Schorff et al., 2015 - FaceNet: A Unified Embedding for Face Recognition and Clustering

Contrastive vs. Triplet Loss

what about WDMC?

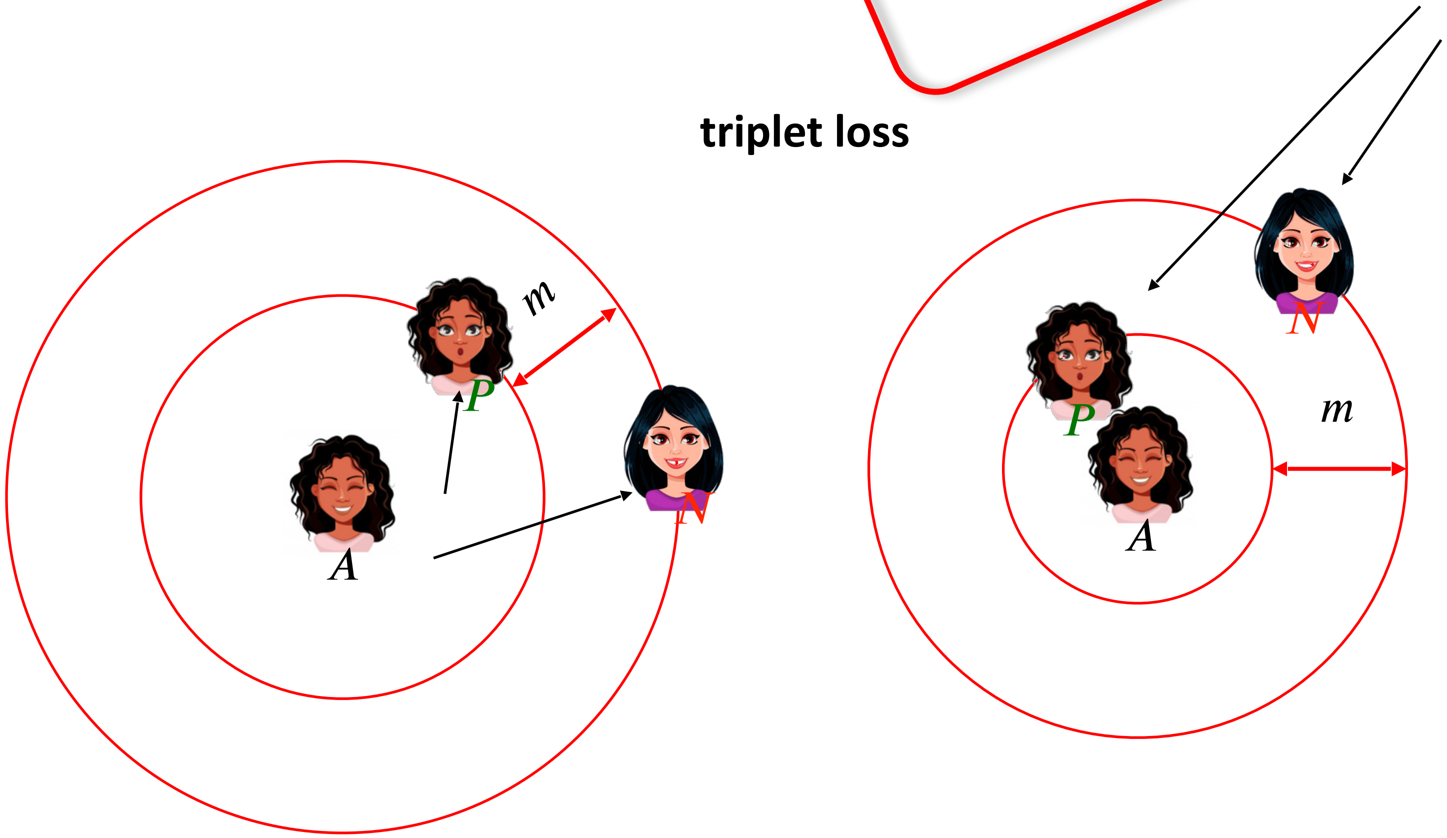
For **a single triplet**:

contrastive loss



f pulls positives closer and pushes negatives away , at least till the margin

triplet loss



f may push away positives

f may pulls negatives closer

provided that the difference between positives and negatives to the anchor is larger than the margin

Triplet loss

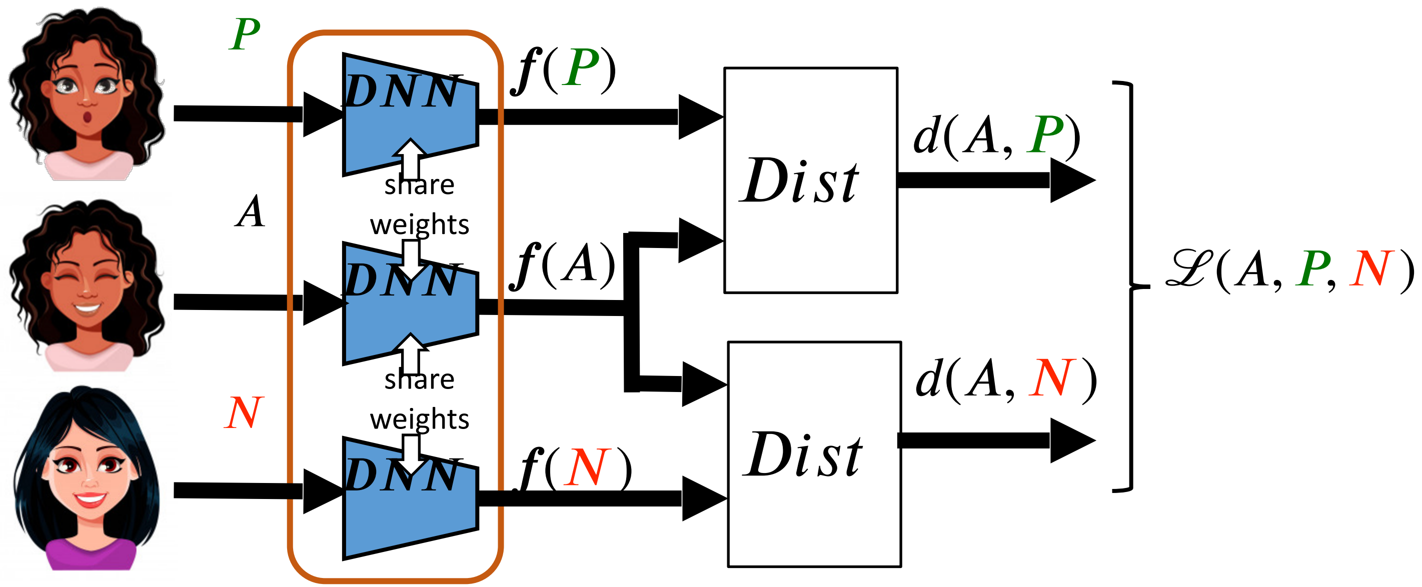
Formulation considering **all triplets** $\{(A_i, P_i, N_i)\}$ in a batch:

$$\boxed{\mathscr{L} = \sum_i max\left(\left(0, \left\| \boldsymbol{f}(A_i) - \boldsymbol{f}(P_i) \right\|^2 - \left\| \boldsymbol{f}(A_i) - \boldsymbol{f}(N_i) \right\|^2 + m\right)\right)}$$

Triplet architecture

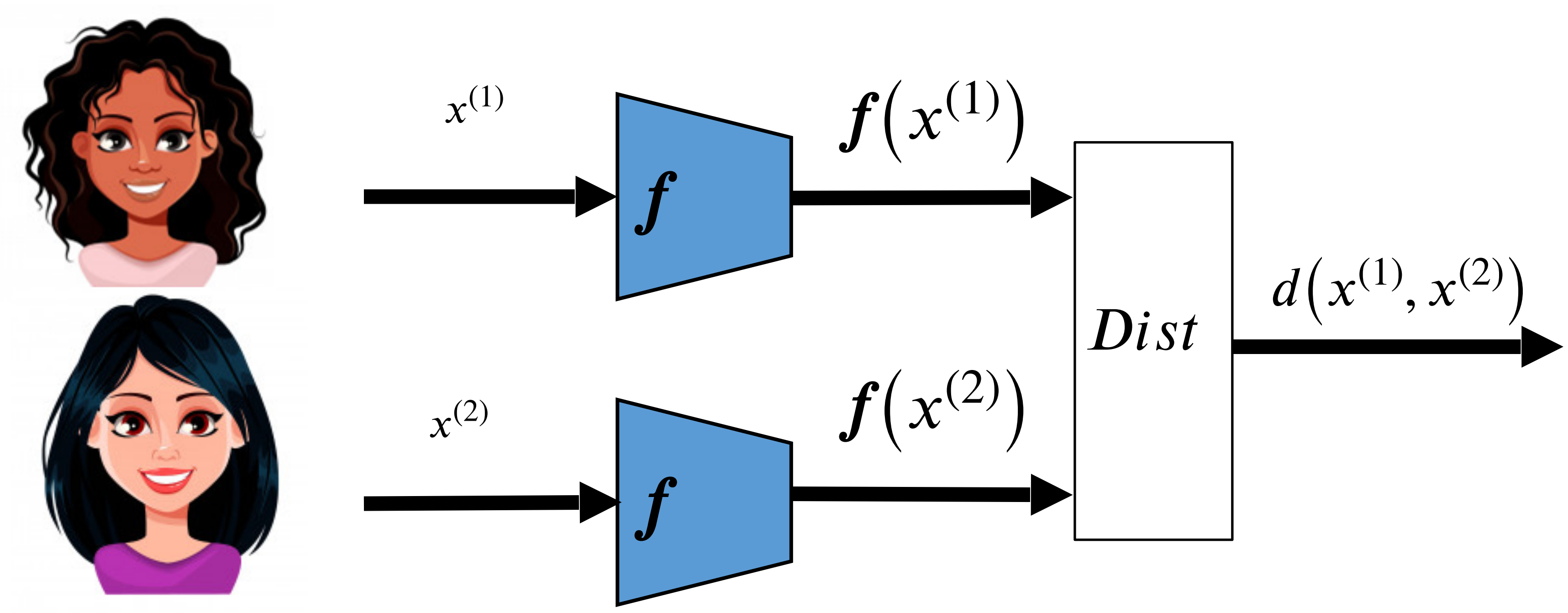
Training

- **3-branch** network
- Positive & negative branches
- Always share weights

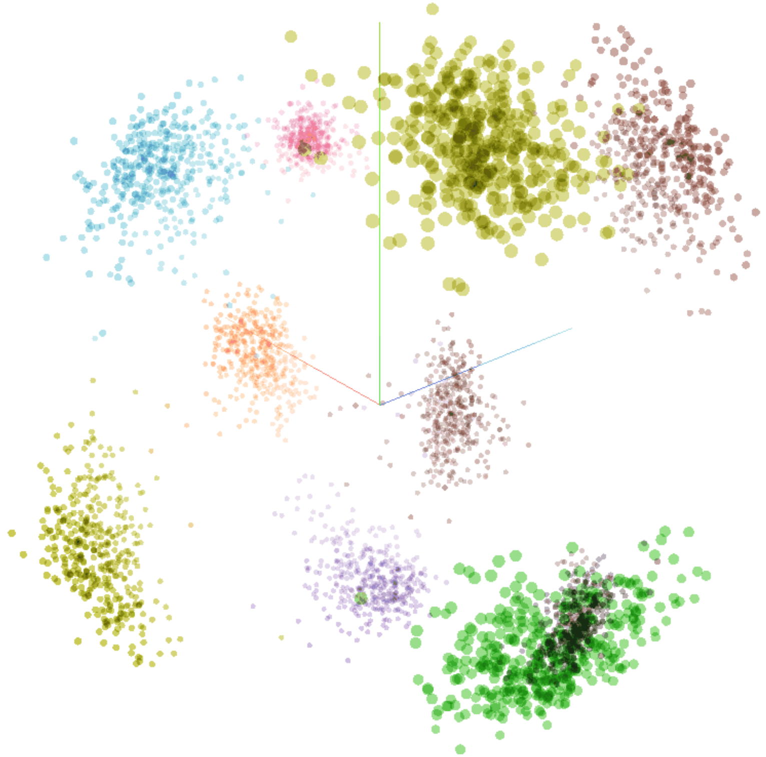


Triplet architecture

Test: same as for the contrastive case.



View of the Embedding Space

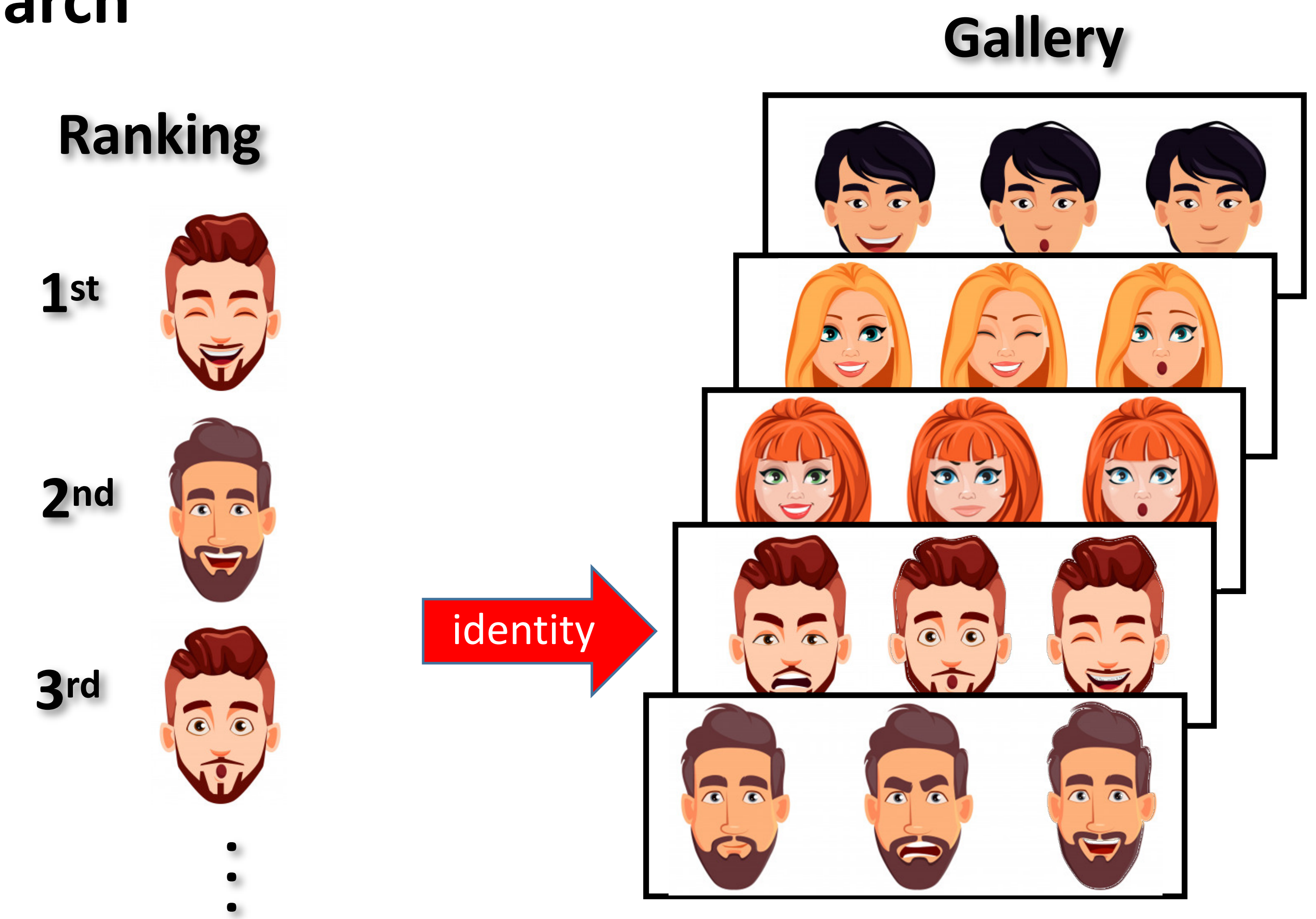


Embedding Space for train data (left) and test data (right) using PCA in Tensorboard

Source: [Enosh Shresta, 2019, "Triplet Loss and Siamesse Networks"]

Test time

Just nearest neighbor search



Often in security applications, the subjects in *n* highest rank positions are analyzed visually by a human expert for a final decision.

Content

Improving Similarity Learning

- Loss:
  - Contrastive vs. triplet loss vs. WDMC
- Sampling
  - Sample the space wisely diversity of classes & hard cases
- Ensembles
  - Multiple networks, each trained with a subset of triplets
- Use the Classification loss for similarity learning

References for Losses

- Wang et al., 2017 Deep metric learning with angular loss.
- Yu et al., 2018, Correcting the triplet selection bias for triplet loss

Improving Similarity Learning

- Loss:
  - Contrastive vs. triplet loss vs. WDMC
- Sampling
  - Sample the space wisely exploiting diversity of classes & hard cases
- Ensembles
  - Multiple networks, each trained with a subset of triplets
- Use the Classification loss for similarity learning

Problems with the Triplet Loss

1. The number of possible triplets is $O(n^3)$. Training with all of them would require a very long time.

2. Recall that

- the triplet loss $\rightarrow$ $\mathscr{L}(A, P, N) = max\left(0, d(A, P)^2 - d(A, N)^2 + m\right)$

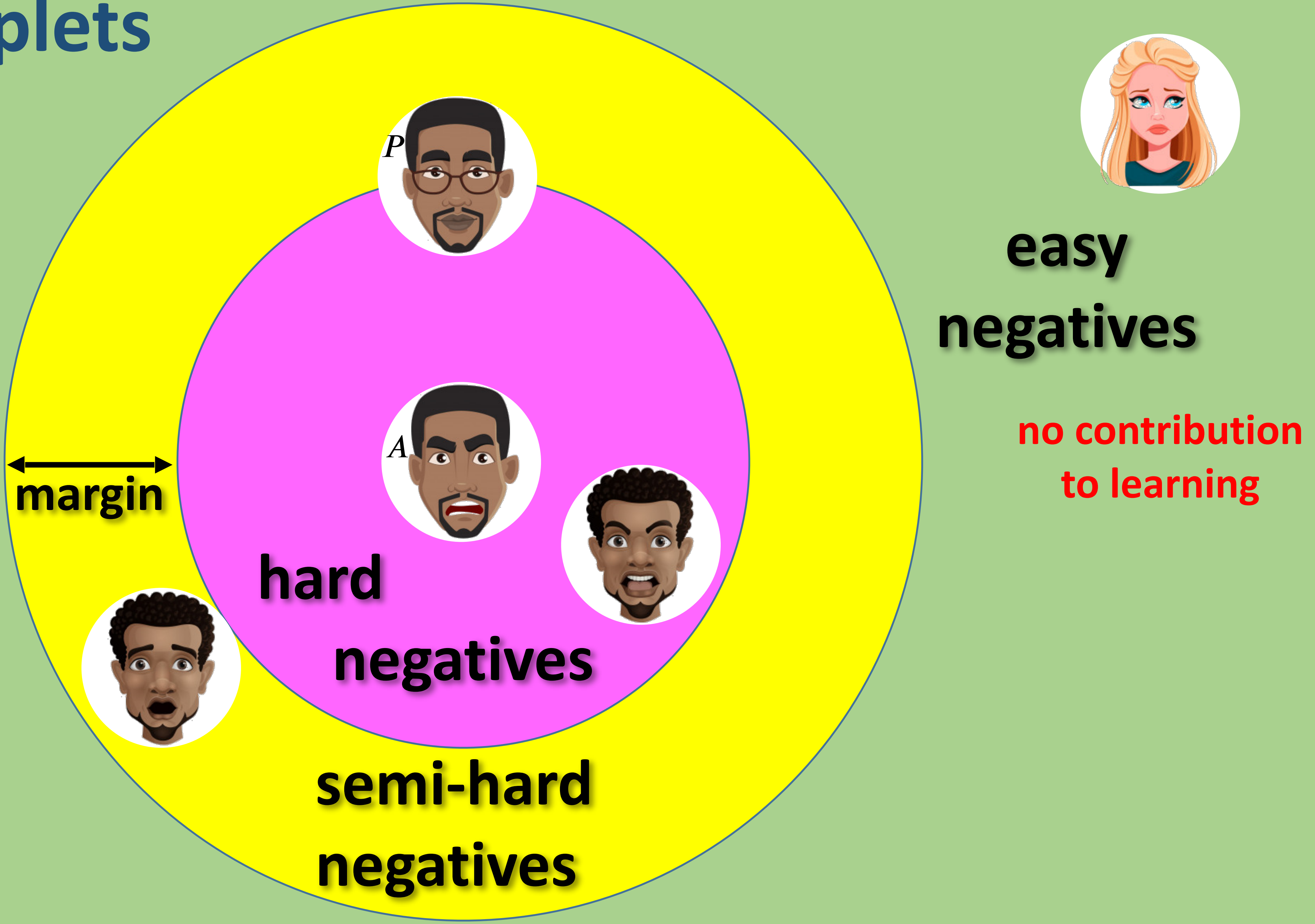
- the gradient of *max* function $\rightarrow$ $\dfrac{d\ max(0,x)}{dx} = \begin{cases} 0 & if\ x \leq 0 \\ 1 & if\ x > 0 \end{cases}$

Therefore, triplets for which

$$\left\| f(A) - f(P) \right\|^2 - \left\| f(A) - f(N) \right\|^2 + m \leq 0$$

will not contribute to parameter learning.

Choosing triplets



Hard triplet mining

General procedure:

choose *initial_triplets* $\{A_i, P_i, N_i\}$,

while *stop_condition* = *false* do

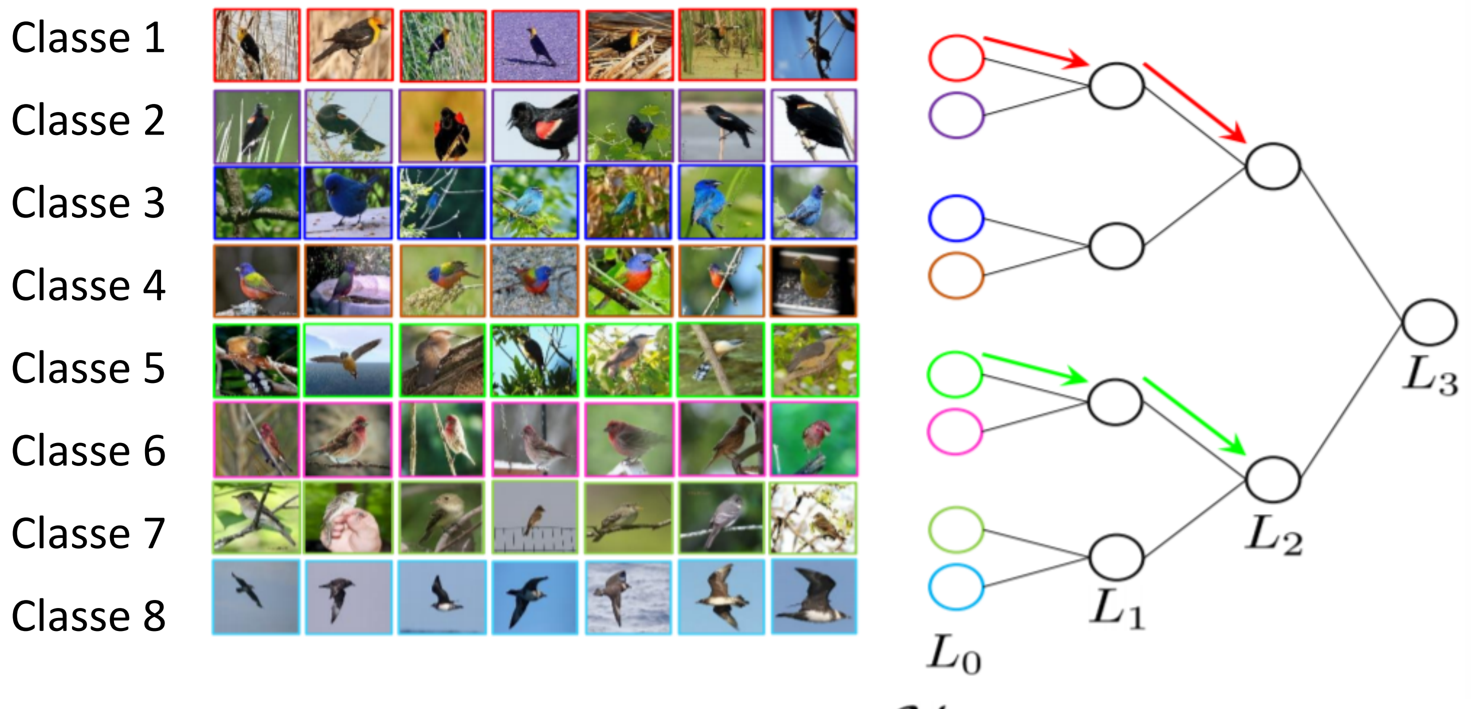
- Train for some epochs
- Choose hard triplets $\{A_i, P_i, N_i\}$

end

Hard triplets can be selected offline (manually) or online (automatically)

Sampling: Hierarchical Triplet Loss

1. Build a hierarchical tree where the leaves represent classes.
2. Recursively merge them until the root node



Hierarchical Tree $\mathcal{H}$

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL: Building the Tree

- To create the tree, first define the **between class distances** $d(p,q)$:

$$d(p,q) = \frac{1}{n_p n_q} \sum_{i \in p, j \in q} \left\| \boldsymbol{r}_i - \boldsymbol{r}_j \right\|^2$$

number of samples in classes p and q

deep features of samples i and j

- If the deep features $\boldsymbol{r}_i$ are normalized into unit length, $d(p,q)$ varies from 0 to 4.

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL: Building the Tree

- Calculate the **average within classes distance** d_0**:**

number of sample in class c

$$d_0 = \frac{1}{\mathscr{C}} \sum_{c=1}^{\mathscr{C}} \left(\underbrace{\frac{1}{n_c^2 - n_c}} \sum_{i,j \in c} \underbrace{\left\| \boldsymbol{r}_i - \boldsymbol{r}_j \right\|^2} \right)$$

number of classes

twice the number of sample pairs in class c

distance between samples i and j in the embedding space

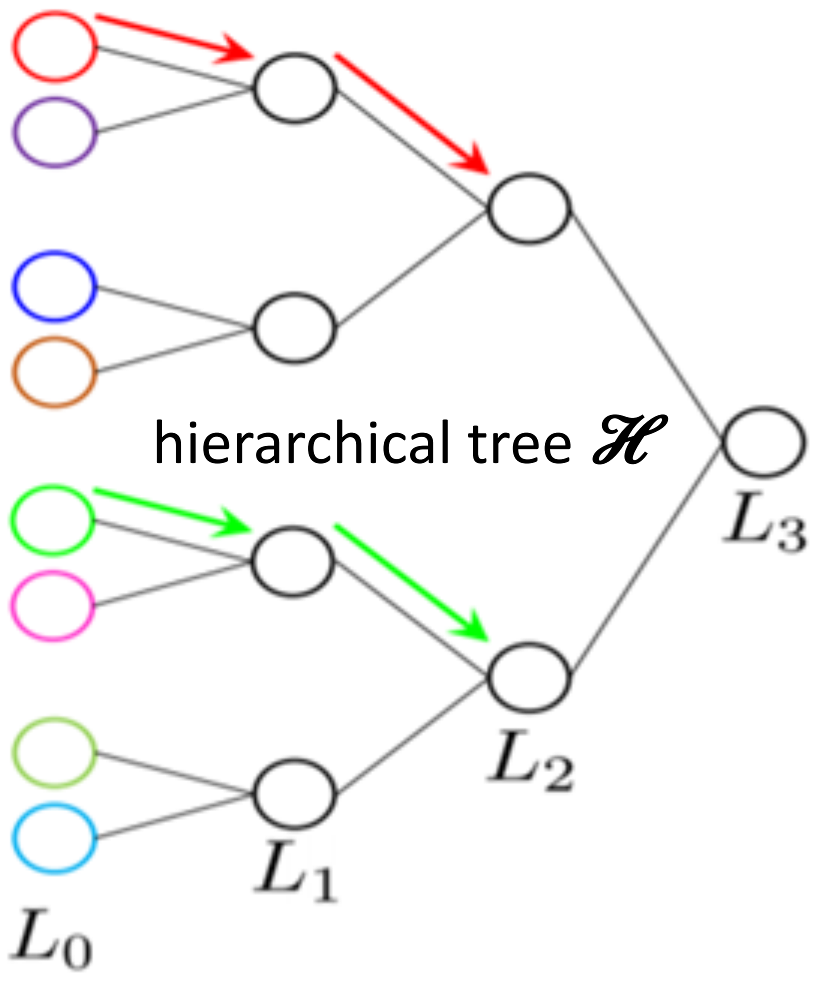
HTL: Building the Tree

- Two classes with a distance less than

$$d_l = \frac{l(4 - d_0)}{L} + d_0$$

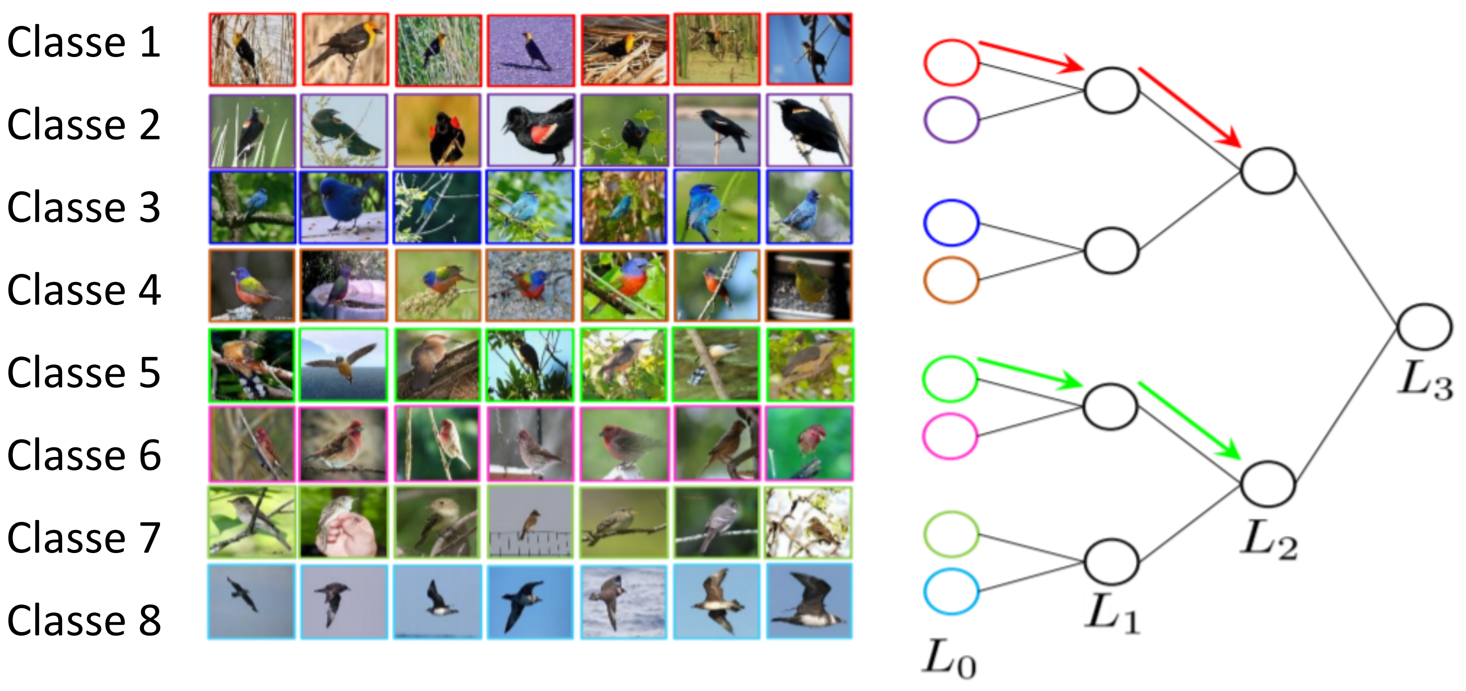
  are merged into a single node at the l-th level.

- Nodes are merged from 0-th level to a single node at to L-th level forming the hierarchical tree $\mathscr{H}$.



HTL: Anchor-Neighbor Sampling

- Randomly select l' nodes at 0th level
  - Preserve class diversity in the mini-batch



Hierarchical Tree $\mathcal{H}$

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL: Anchor-Neighbor Sampling

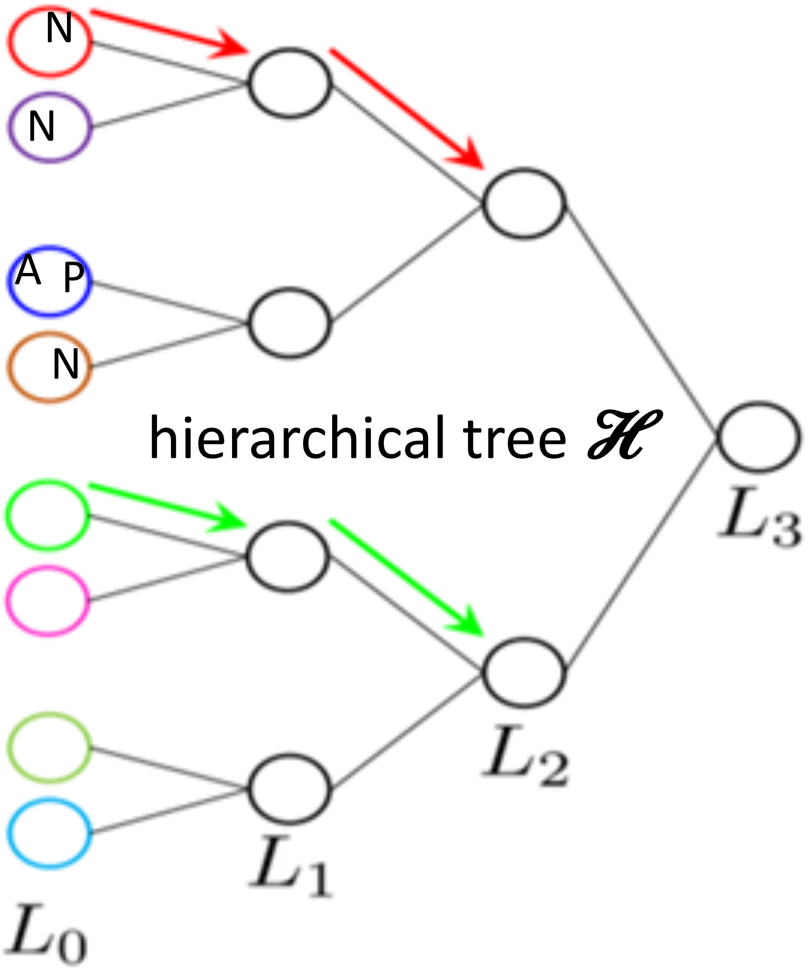
- Randomly select l' nodes at 0th level, and select one **anchor** sample from each node
- For each anchor select the furthest **positive** sample.
- For each of the l' anchors, pick the k nearest nodes/classes at 0th level based on the class distance measured in the feature space.
- Collect t samples from these nodes/classes randomly to be **negative** samples.

We will end up with a mini batch containing $l'\ k\ t$ triplets.

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL: Anchor-Neighbor Sampling

- Randomly select l' nodes at 0th level, and select one **anchor** sample from each node
- For each anchor select the furthest **positive** sample.
- For each of the l' anchors, pick the k nearest nodes/ classes at 0th level based on the class distance measured in the feature space.
- Collect t samples from these nodes/classes randomly to be **negative** samples.



We will end up with a mini batch containing l' k t triplets.

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL: Loss Formulation

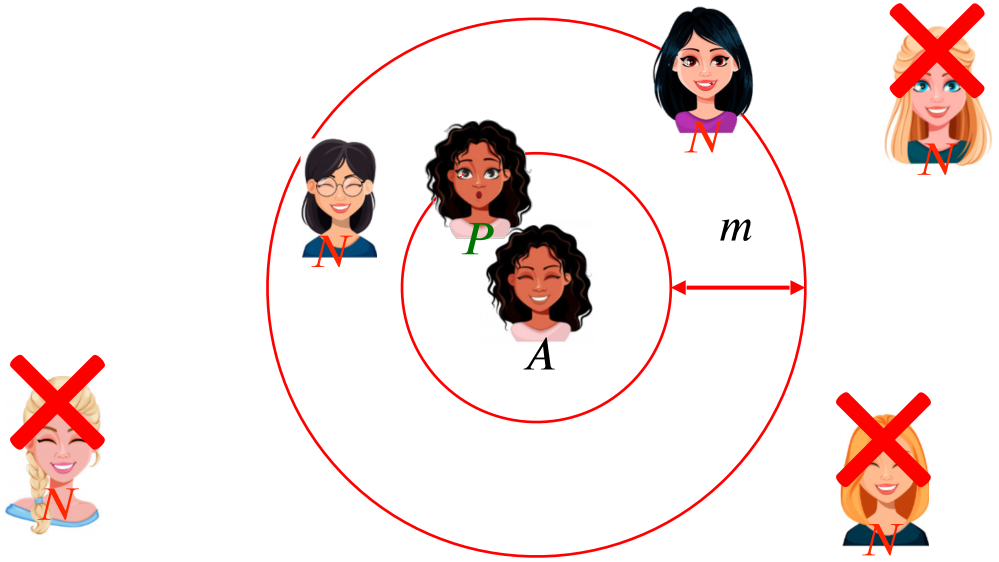
$$\mathscr{L} = \sum_i max\left(0, \left\| \boldsymbol{f}(\boldsymbol{x}_A^i) - \boldsymbol{f}(\boldsymbol{x}_P^i) \right\|^2 - \left\| (\boldsymbol{x}_A^i) - \boldsymbol{f}(\boldsymbol{x}_N^i) \right\|^2 + m \right)$$

all the triplets in the batch

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

Dynamic Margin

- Hard negative mining rejects negatives “too far” from the anchor, whereby “too far” is defined by the margin m.

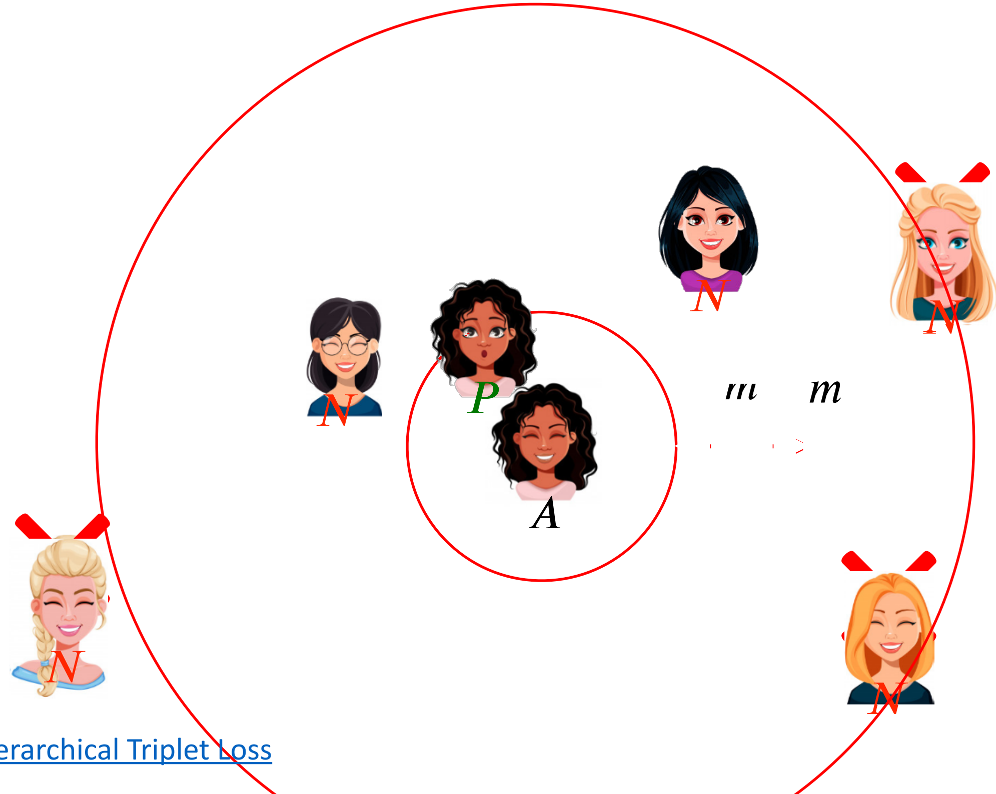


waste of samples!

[Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss]

Dynamic Margin

- *"… to learn from more meaningful hard samples with the guide of global context,"* we keep the distant negatives and enlarge the margin.



[Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss](#)

Dynamic Margin

… the loss takes the form

$$\mathscr{L} = \sum_i max\left(0, \left\| \boldsymbol{f}\left(\boldsymbol{x}_A^i\right) - \boldsymbol{f}\left(\boldsymbol{x}_P^i\right) \right\|^2 - \left\| \left(\boldsymbol{x}_A^i\right) - \boldsymbol{f}\left(\boldsymbol{x}_N^i\right) \right\|^2 + m_i \right)$$

The margin m_i depends on the distances between the anchor and the negative classes computed in the hierarchical tree (see ref. for details).

The distances change as training progresses. Therefore, the tree must be recalculated after some epochs.

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL Performance

Example:

Results on the *In-Shop Clothes Retrieval* *[dataset]*

The accuracy metric is the ***Recall@K***.

"If one of the K retrieved images has the same label as the query, the recall will increase by 1."

Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss

HTL Performance

Accuracy

Comparisons on the In-Shop Clothes Retrieval dataset

	R@	1	10	20	30	40	50
	FashionNet+Joints[15]	41.0	64.0	68.0	71.0	73.0	73.5
	FashionNet+Poselets[15]	42.0	65.0	70.0	72.0	72.0	75.0
	FashionNet[15]	53.0	73.0	76.0	77.0	79.0	80.0
	HDC[38]	62.1	84.9	89.0	91.2	92.3	93.1
	BIER[18]	76.9	92.8	95.2	96.2	96.7	97.1
Triplet loss →	Ours Baseline	62.3	85.1	89.0	91.1	92.4	93.4
Anchor-Neighbor →	A-N Sampling	75.3	91.8	94.3	96.2	96.7	97.5
With dynamic margin →	HTL	**80.9**	**94.3**	**95.8**	**97.2**	**97.4**	**97.8**

[Ge et al., 2018, Deep Metric Learning with Hierarchical Triplet Loss]

References for Sampling

- Manmatha et al., 2017, Sampling matters for deep metric learning.
- Xu et al., 2019, Deep asymmetric learning via rich relationship mining.
- Duan et al., 2019, Deep embedding learning with discriminative sampling policy
- Wang et al., 2019, Multi-similarity loss with general pair weighting for deep metric learning.
- Wang et al., 2019, Ranked list loss for deep metric learning.

Improving Similarity Learning

- Loss:
  - Contrastive vs. triplet loss vs. WDMC
- Sampling
  - Sample the space wisely diversity of classes & hard cases
- Ensembles
  - Multiple networks, each trained with a subset of triplets
- Use the Classification loss for similarity learning

Ensembles: Divide and Conquer

Rational:

- The goal is learning an embedding space where semantically similar objects are close and dissimilar objects are far apart.
- We have been after a single distance metric that encodes many different notions of similarity such as color, shape, pose or semantic meaning. This is too much for a single NN.

Intuition:

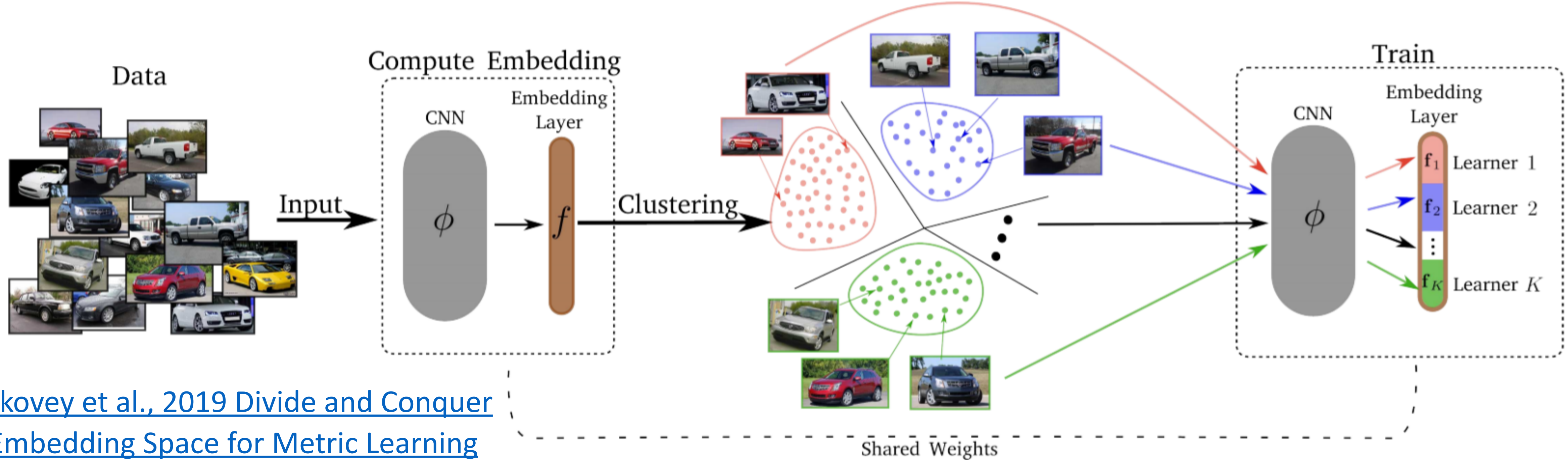
- Split the problem into simpler sub-problems, train the learners separately, and merge them after they have converged.

Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning

Divide & Conquer

The mapping into the d-dimensional embedding space is a non linear function ϕ followed by a linear layer f.

So, representation of an input $\boldsymbol{x}$ in the embedding space will be the d-dimensional vector $f\left(\phi(\boldsymbol{x})\right)$

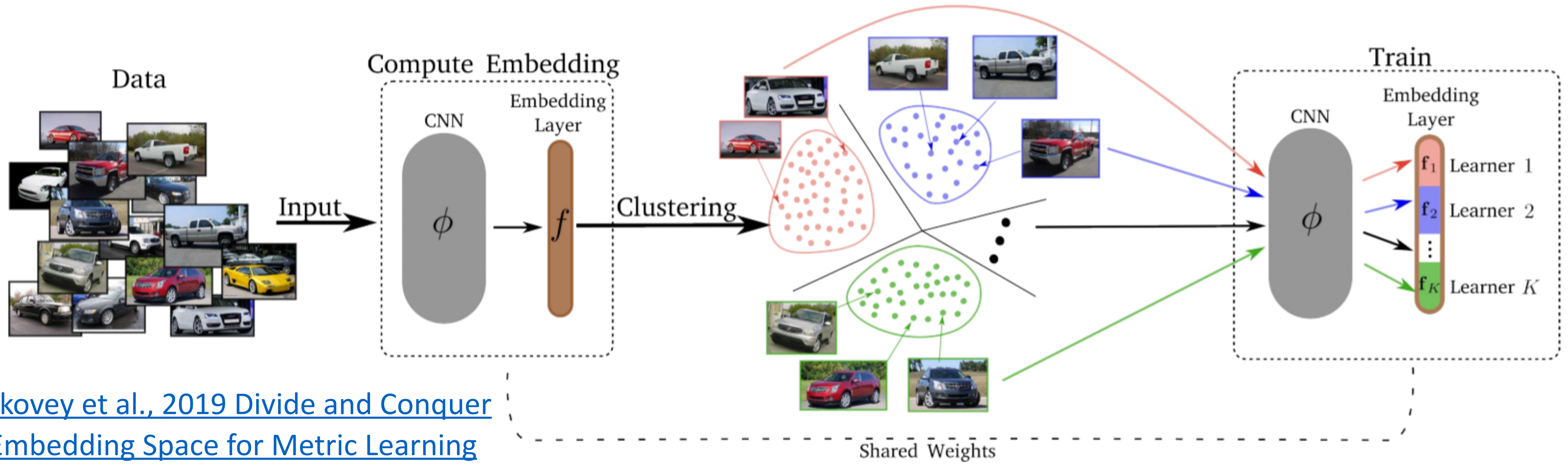


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]

Divide & Conquer

DIVIDE

1) Cluster the samples in the embedding space into *K* cluster using K-means

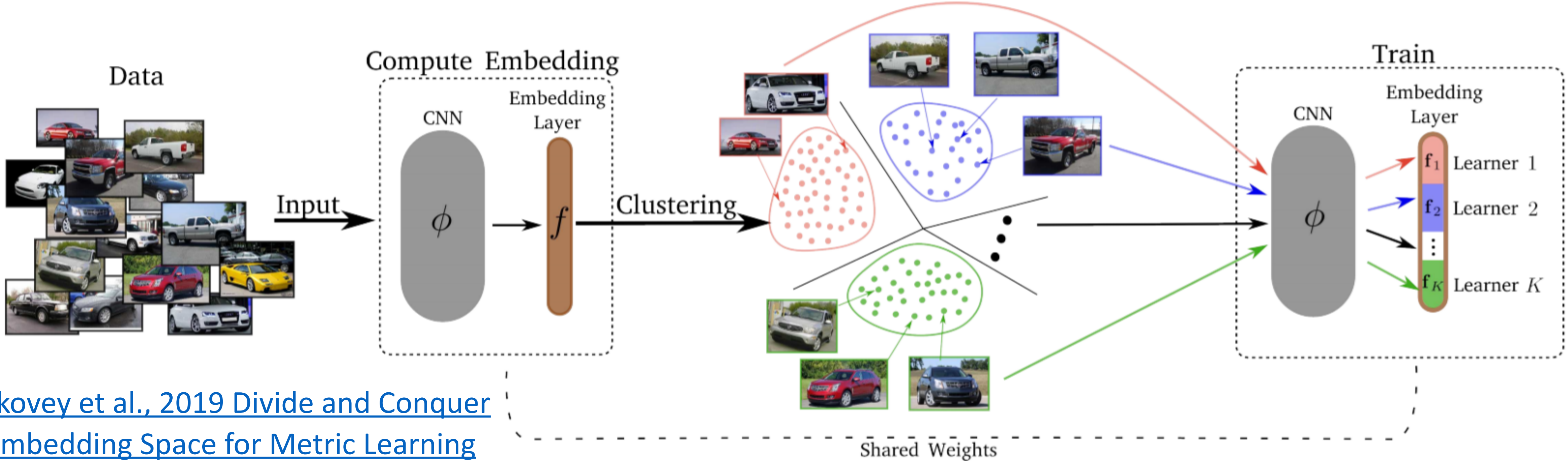


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]()

Divide & Conquer

DIVIDE

2) Define K learners consisting of the same non linear function ϕ followed by a linear layer f_k, that maps the input into a d/K dimensional space

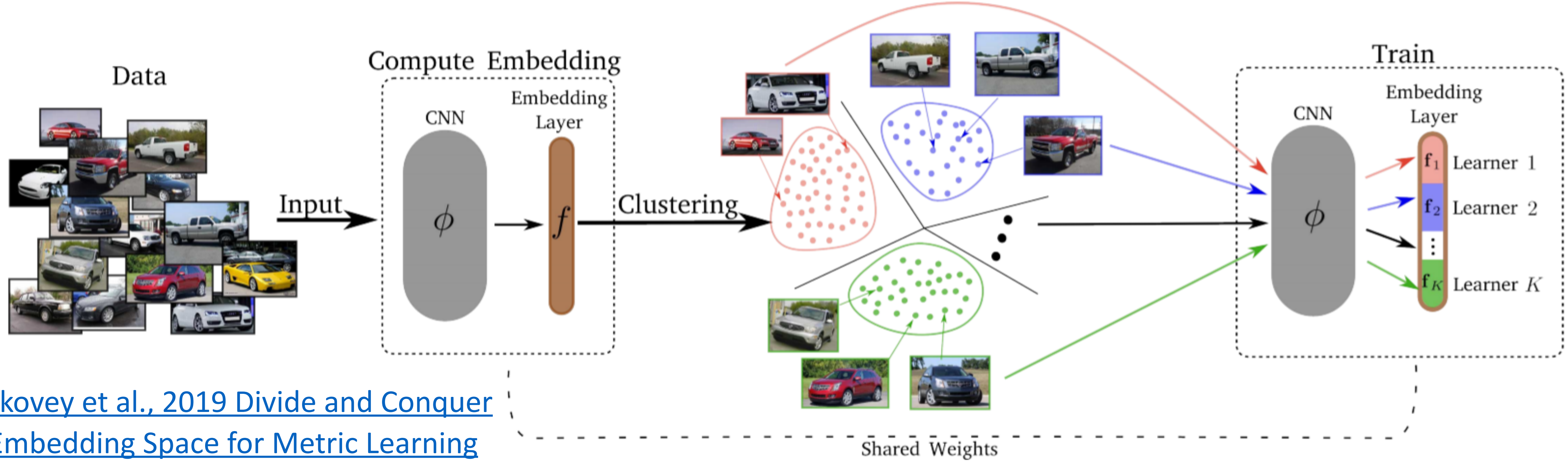


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]()

Divide & Conquer

DIVIDE

3) Train alternately the K learners with samples from the k-th cluster. Note that each backward pass will update the parameters of the shared representation ϕ and the parameters of the current learner f_k.

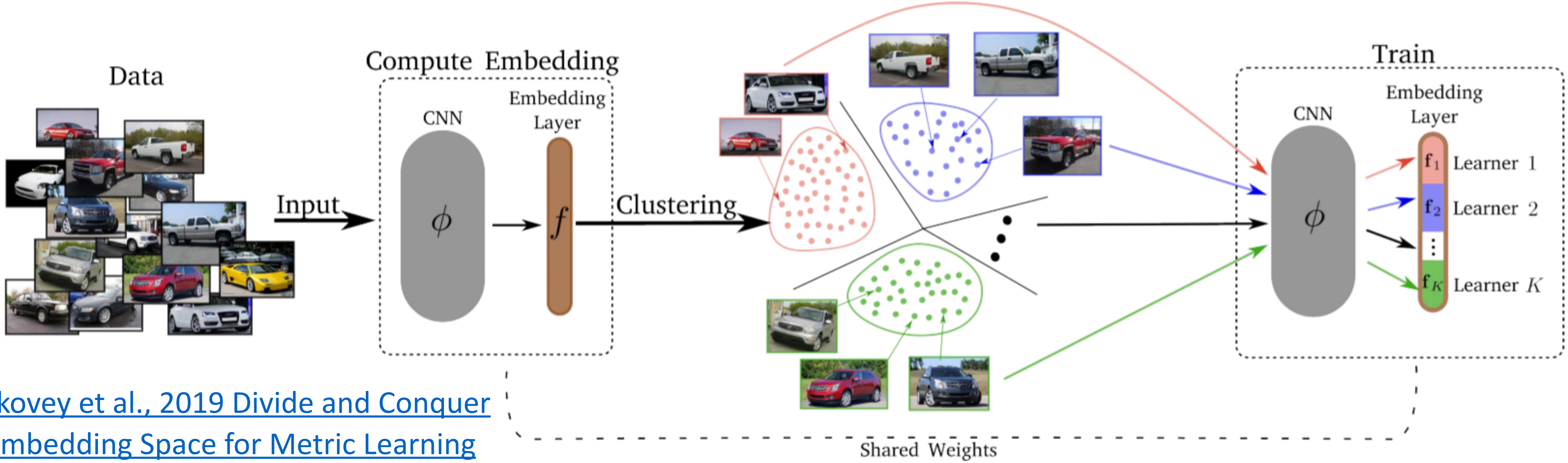


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]

Divide & Conquer

DIVIDE

4) Every T epochs re-cluster using the full embedding's

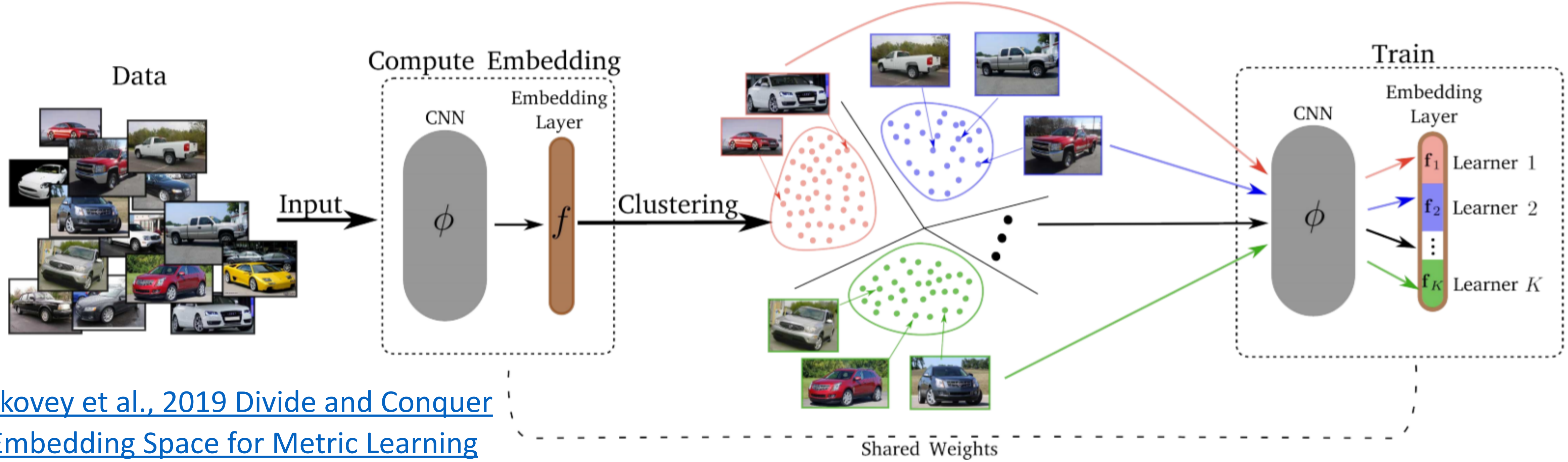


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]

Divide & Conquer

CONQUER

5) After convergence, merge their solution to get the full embedding space back

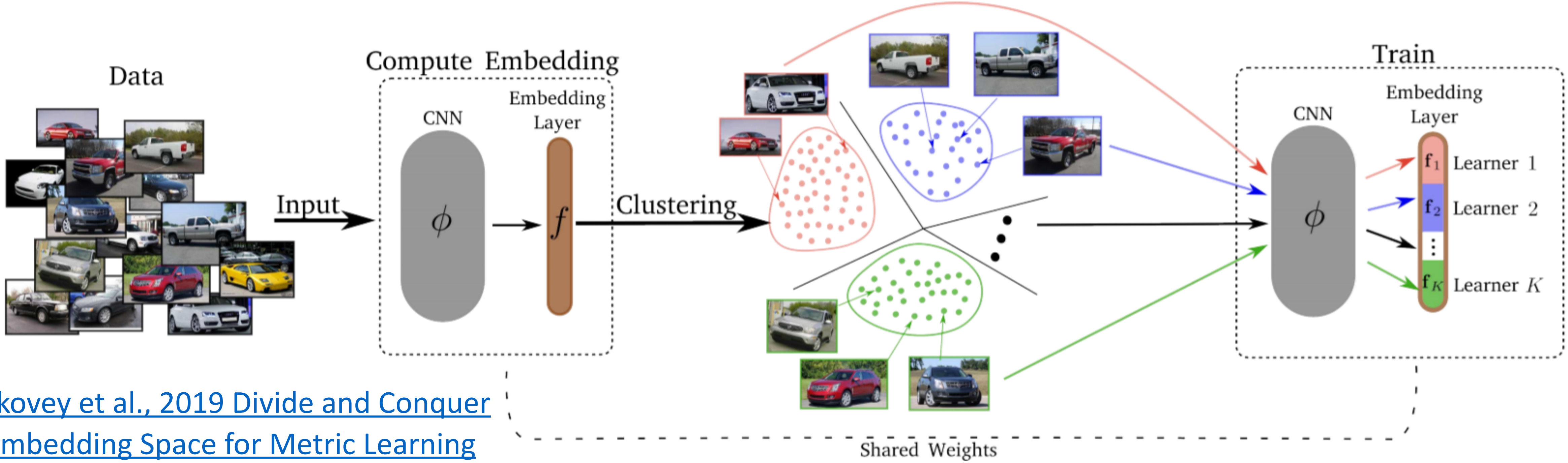


Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning

Divide & Conquer

CONQUER

6) Fine tune the embedding layer on the entire data set to achieve consistency between embedding of the individual learners.

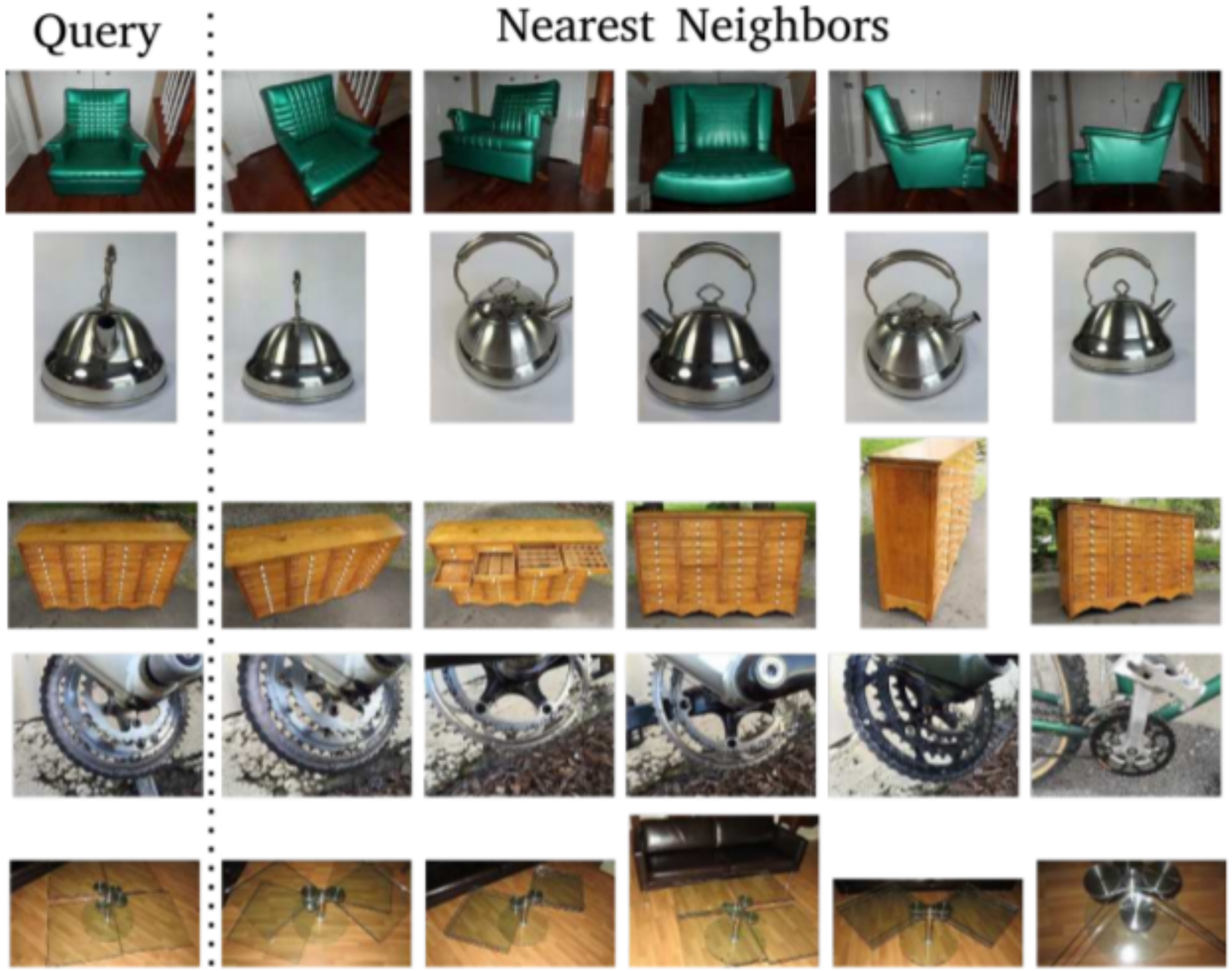


[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]()

Divide & Conquer Performance

Example:

Results on the n *Stanford Online Product* [*dataset*]



[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]

Divide & Conquer Performance

Example:

Results on the n *Stanford Online Product* *[dataset]*

HTL →

Divide & Conquer→

R@k	1	2	4	8	NMI
Triplet Semihard [42]	51.5	63.8	73.5	82.4	53.4
LiftedStruct [32]	48.3	61.1	71.8	81.1	55.1
FacilityLoc [42]	58.1	70.6	80.3	87.8	59.0
SmartMining [14]	64.7	76.2	84.2	90.2	-
N-pairs [41]	71.1	79.7	86.5	91.6	64.0
Angular [47]	71.4	81.4	87.5	92.1	63.2
ProxyNCA [30]	73.2	82.4	86.4	88.7	64.9
HDC [51]	73.7	83.2	89.5	93.8	-
DAML (N-pairs) [6]	75.1	83.8	89.7	93.5	66.0
HTG [53]	76.5	84.7	90.4	94	-
BIER [33]	78.0	85.8	91.1	95.1	-
HTL [10]	81.4	88.0	92.7	95.7	-
DVML [28]	82.0	88.4	93.3	96.3	67.6
A-BIER [34]	82.0	89.0	93.2	96.1	-
Margin baseline [49]	79.6	86.5	91.9	95.1	69.1
Ours (Margin)	**84.6**	**90.7**	**94.1**	**96.5**	**70.3**
DREML [50]	86.0	91.7	95.0	97.2	76.4

Table 2: Recall@k for $k = 1, 2, 4, 8$ and NMI on CARS196

[Sanakovey et al., 2019 Divide and Conquer the Embedding Space for Metric Learning]

References for Ensembles

- Opitz et al., 2017, Boosting Independent Embeddings Robustly
- Elezi et al., 2020, The Group Loss for Metric Learning
- Yuan et al., 2017, Hard-Aware Deeply Cascaded Embedding
- Wang et al., 2019, Ranked list loss for deep metric learning

Improving Similarity Learning

- Loss:
  - Contrastive vs. triplet loss vs. WDMC
- Sampling
  - Sample the space wisely diversity of classes & hard cases
- Ensembles
  - Multiple networks, each trained with a subset of triplets
- Use the Classification loss for similarity learning

References for Classification loss

- Movshovitz-Attlas et al., 2017, No Fuss Distance Metric Learning using Proxies
- Teh et al., 2020, ProxyNCA++L Revisiting and Revitalizing Proxy Neighborhood Component Analysis
- Qian et al, 2019, SoftTriple Loss: Deep Metric Learning Without Triplet Sampling
- Elezi et al., 2020, The Group Loss for Deep Metric Learning.

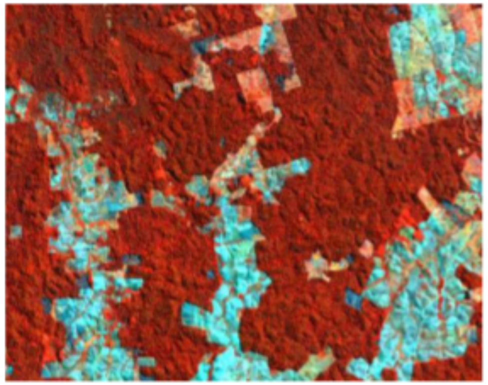
Content

- One-shot Learning
- Example: face recognition
- Similarity Learning
- Siamese Networks
- Improving Similarity Learning
- Change Detection

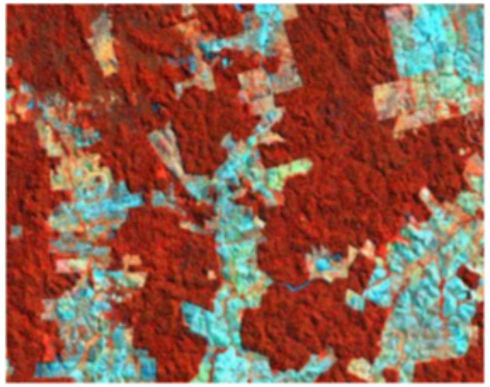
Change Detection

Change detection involves finding the changes in images of the same location taken at different times.

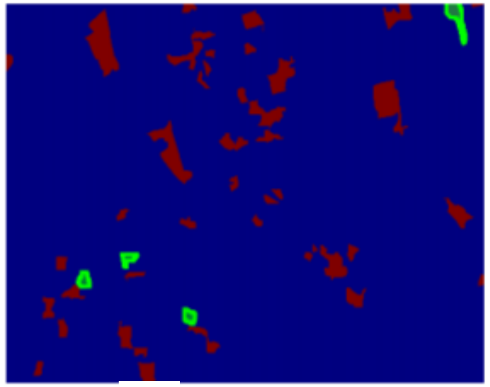
Example: Deforestation in Brazilian Amazon and Cerrado Biomes*



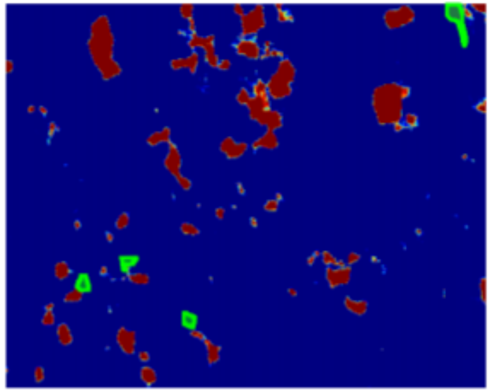
T1 (2016)



T2 (2017)



Reference



SN-4T

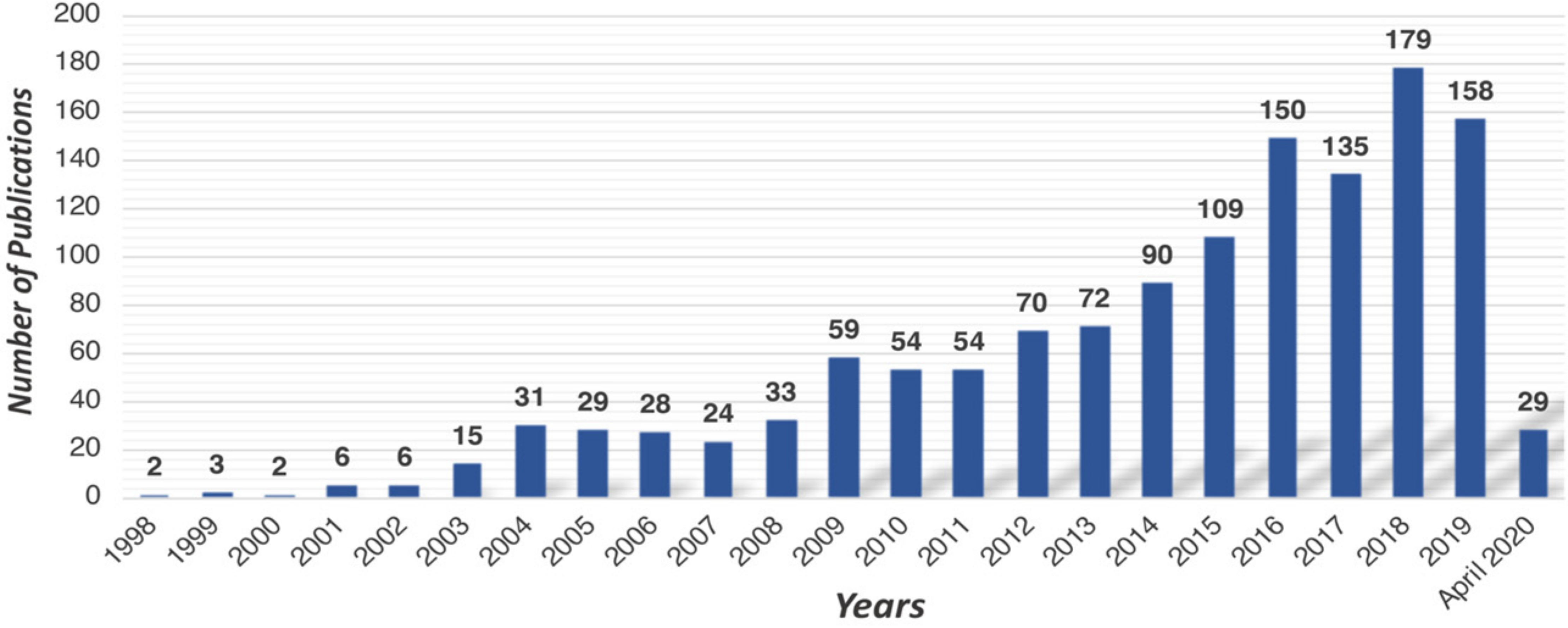
*Source: [Adarme et al.. 2020, “Evaluation of Deep Learning Techniques for Deforestation Detection in the Brazilian Amazon and Cerrado Biomes From Remote Sensing Imagery”]

Change Detection Applications

- land use and land cover change (Liu et al. 2018a, b, c, d, e) (Amici et al. 2017),
- deforestation (Adarme et al., 2020),
- urban settlements (Kleynhans et al. 2015) ,
- changes as part of natural calamities (Hölbling et al. 2015) (Feizizadeh et al. 2017),
- …

Research on Change Detection

Published literature on urban change detection, according to the keywords *remote sensing* and *urban change detection*.



Source: [You, Cao and Zhou, 2020, "A Survey of Change Detection Methods Based on Remote Sensing Images for Multi-Source and Multi-Objective Scenarios"]

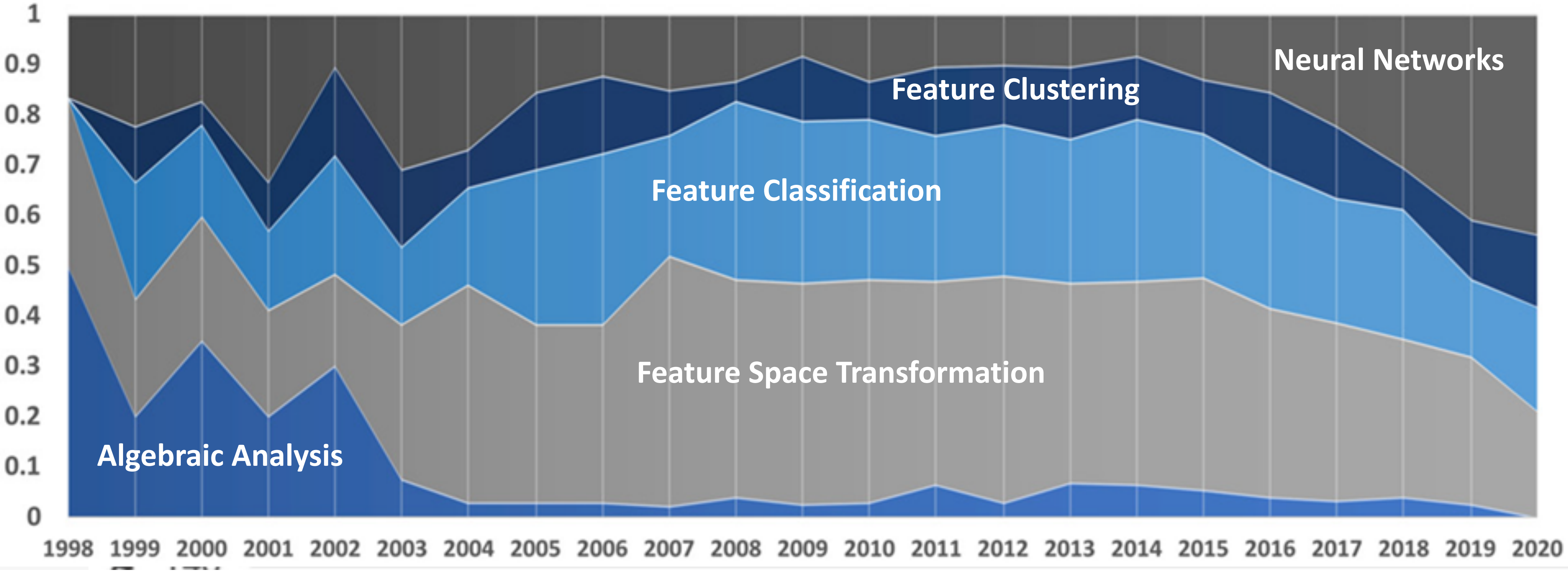
Change Detection Databases

#	Database	Weblink	Resolution	No: of images
1.	Landsat 5 and Landsat 8 database (Zanchetta et al. 2016), (Kant and Singh 2017)	http://glovis.usgs.gov/	2634 × 3126	42,992
2.	PALSAR mosaic Synthetic Aperture Radar (SAR), Sentinel C band SAR and optical data (Johnson et al. 2017)	http://www.eorc.jaxa.jp/ALOS/en/palsar_fnf/fnf_index.htm, https://scihub.copernicus.eu/	4500 × 4500	108
3.	MODIS–VCF(MOD44B), Landsat dataset (Amarnath et al. 2017)	http://glcf.umiacs.umd.edu/data/vcf/	6406 × 5521	500
4.	Landsat database (Haque and Basak 2017), (Uchenna et al. 2017)	www.earthexplorer.usgs.gov	938 × 528	Daily image updation
5.	Landsat database (Kaliraj et al. 2017)	http://glcfapp.glcf.umd.edu:8080/esdi/	6406 × 5521	800
6.	Moderate-resolution Imaging Spectroradiometer (MODIS) surface reflectance products (MOD09A1) (Qiu et al. 2017)	https://ladsweb.nascom.nasa.gov/	600 × 483	Daily image updation
7.	Landsat surface reflectance images (Ye et al. 2018)	http://earth.google.com/	4096 × 4096	Daily image updation
8.	Pleiades imagery (Suresh and Lal 2017a)	http://www.satpalda.com/ gallery/pleiades-imagery/	1464 × 1143	Daily image updation
9.	IR imagery and HURDAT2 data (Pradhan et al. 2018)	http://www.nrlmry.navy.mil and http://www.nhc.noaa.gov/data/#hurdat	625 × 625	200
10.	Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor data (R et al. 2017)	http://aviris.jpl.nasa.gov/	677 × 512	4000
11.	World-View2 Multispectral and Panchromatic imagery (Luo et al. 2016)	https://www.digitalglobe.com/	1024 × 614	5000
12.	Global land cover dataset (Wang et al. 2018a, b)	http://www.globallandcover.com/	700 × 400	853
13.	Landsat images (Pandey and Khare 2017)	http://bhuvan.nrsc.gov.in	931 × 644	Daily updation

Source: Asokan and Anitha, 2019, “Change detection techniques for remote sensing applications: a survey”

Methods for Change Detection

Deep Learning-based approaches are gaining prominence.



*: [You, Cao and Zhou, 2020, "A Survey of Change Detection Methods Based on Remote Sensing Images for Multi-Source and Multi-Objective Scenarios"]

Interesting Recent Publications

- You, Cao and Zhou, 2020, A Survey of Change Detection Methods Based on Remote Sensing Images for Multi-Source and Multi-Objective Scenarios
- Asokan & Jude, 2019, Change detection techniques for remote sensing applications: a survey, Earth Science Informatics,
- Xudong, et al., 2019, A Conditional Adversarial Network for Change Detection in Heterogeneous Image, IEEE Geoscience and Remote Sensing Letter
- Chen et al., 2020, DASNet: Dual attentive fully convolutional siamese networks for change detection of high resolution satellite images
- Touati, Mignotte, and Dahmane, 2020, Partly Uncoupled Siamese Model for Change Detection from Heterogeneous Remote Sensing Imagery, Journal of Remote Sensing & GIS
- Zhang et al., 2019, Triplet-Based Semantic Relation Learning for Aerial Remote Sensing Image Change Detection, IEEE Geoscience and Remote Sensing Letters,

Next Lecture

Tuesday
Lab + 4th Programming Assignment

Similarity Learning

See you next class! ☺