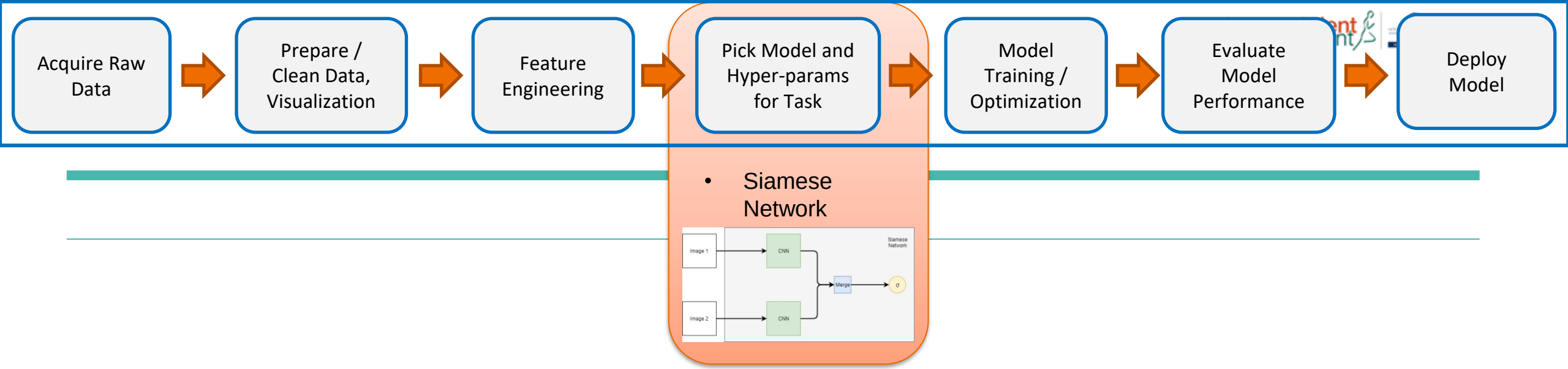
* choice of BD/Arch

* Training — BP+++

Learning Settings

* (Data/Amnt), LossFn, Learn Sett



Siamese and Loss Functions

Recognition Vs Verification

- Recognition: eg. K–Class recognition
  - Given a sample, decide which one of the K it is

- Verification: Given a sample and a class ID,
  - Say “YES” or “NO”

- What makes these two different?
  - Popular in many situations (Eg. Biometrics)
    - Face, Fingerprint
  - An Intra-class variation could be higher than inter-class
    - People change appearance over time.

Verification: Test/verification time

- Often done with nearest neighbors
  - Too much of computing?

- Find “K” nearest neighbors Find the distance to the samples from the class under question
  - Find Mean distance
  - Find Min Distance

- Variable Threshold: Subject specific
  - Some people have high variability in “signature”

Verification: Enrolment/Training

- Usually only small number of samples to add
  - One shot learning (single sample setting)

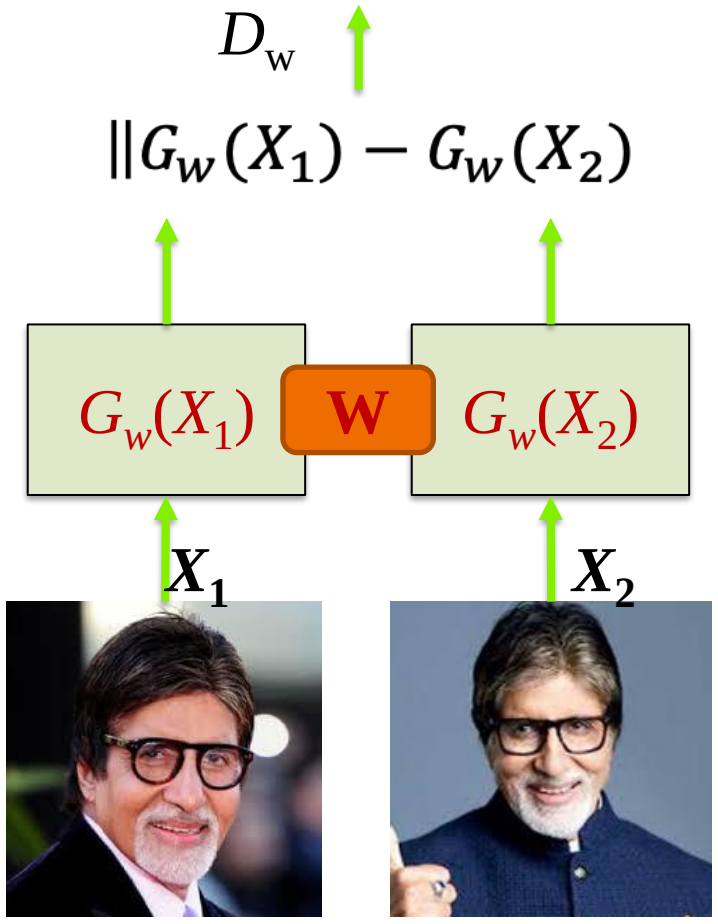
- Find a feature space that suits the local “geometry”
  - Metric learning
  - Aka finding an appropriate distance function

Points to Notice

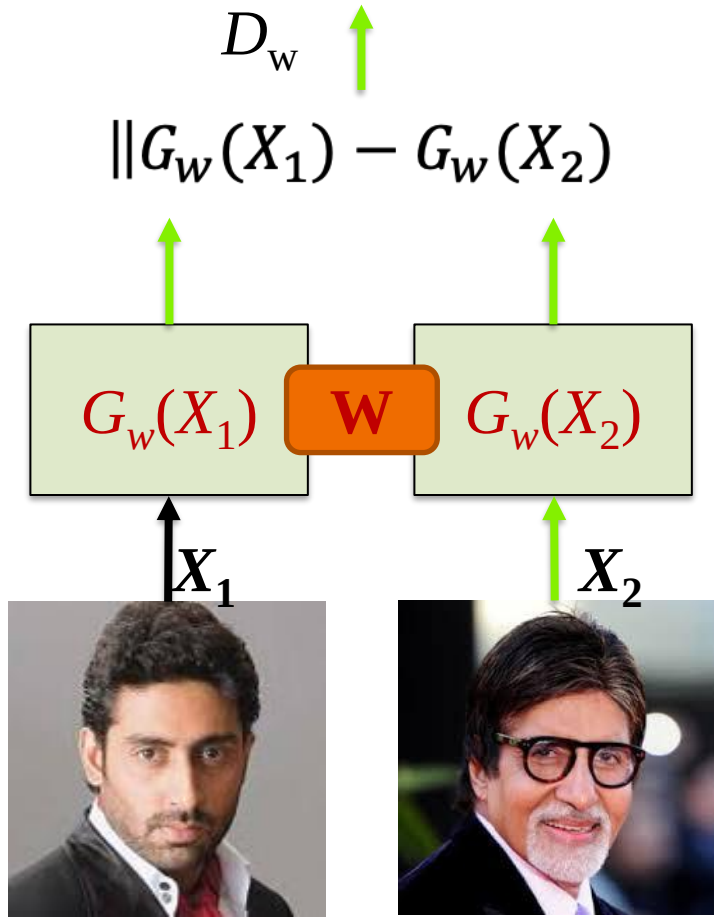
- Features that can capture “Fine Variation”
- Classification Vs Verification
- Learning with a “different style” of Supervision
- Contrastive Learning and Self Supervised Learning

Siamese Architecture/Loss

Make this smaller



Make this larger



- Only pair-wise Labels
- Similarity Metric:
  $D_w(X_1, X_2)$
- Have shared weights
- Training in batches

Historic

- **Input**: A pair of input signatures.
- **Output (Target)**: A label, 0 for **similar,** 1 **else**.

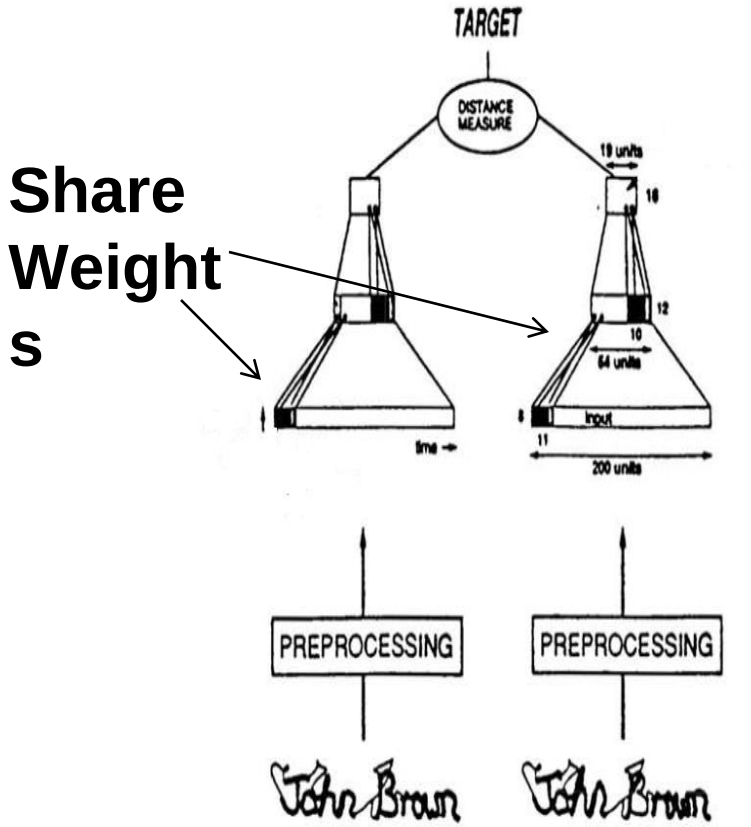
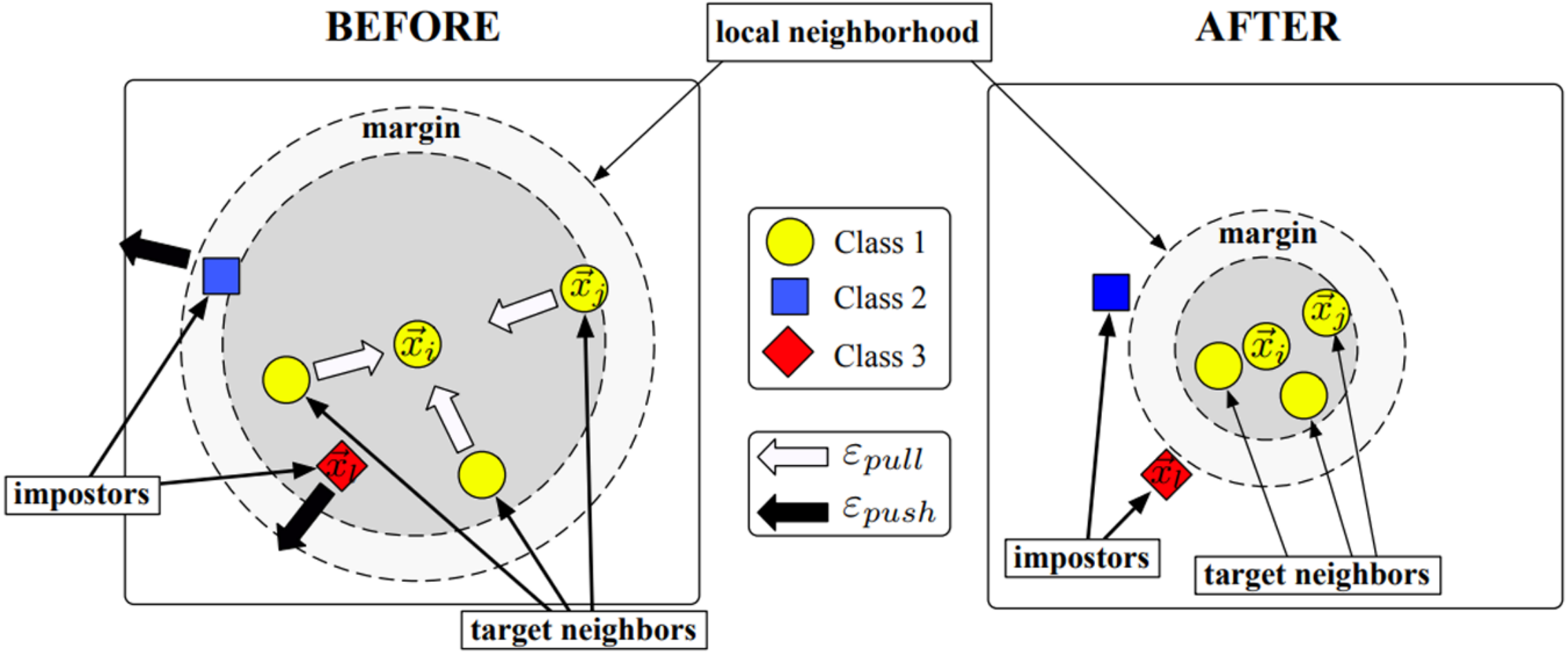


Image Source: Google

Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E. and Shah, R., 1993. Signature Verification Using A "Siamese" Time Delay Neural Network. *IJPRAI*, *7*(4), pp.669-688.

Historic: Large Margin Nearest Neighbour(LMNN)



Intuition in Nearest Neighbor Setting

Weinberger and Saul, JMLR 2009

Blank Slide

Downstream task.

* Task / Goal

BB

Input

output

Video

DDP

100P

[] [] -> Y/N

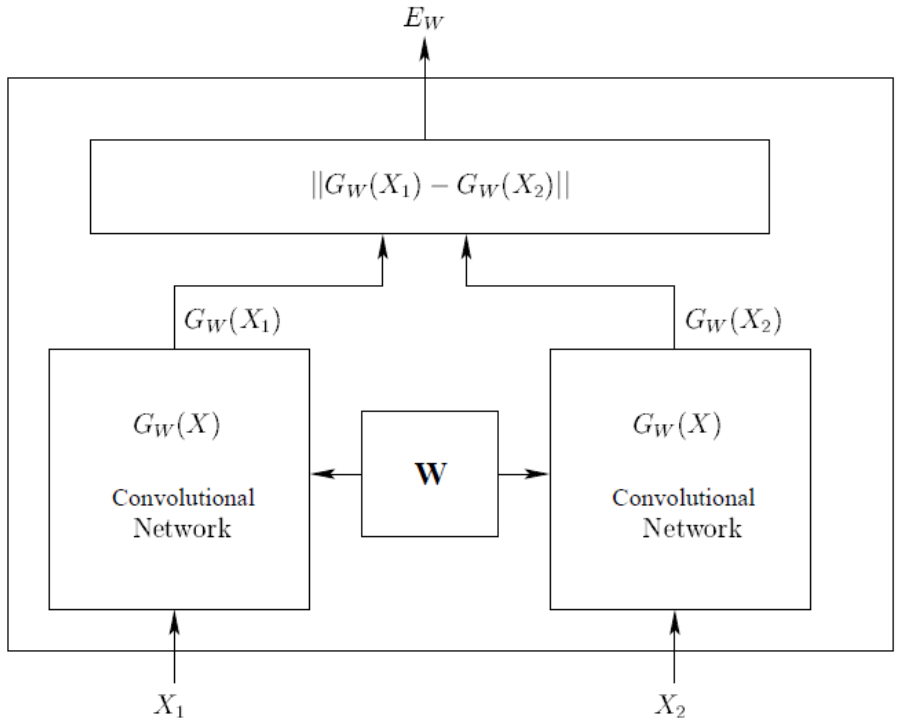
[] [] -> Y/N

R.P N.P

Siamese Architecture

- Given a family of functions $G_W(X)$ parameterized by W, find W such that the similarity metric $D_W(X_1, X_2)$ is small for similar pairs and large for dissimilar pairs:-

$$D_W(X_1, X_2) = \|G_W(X_1) - G_W(X_2)\|$$

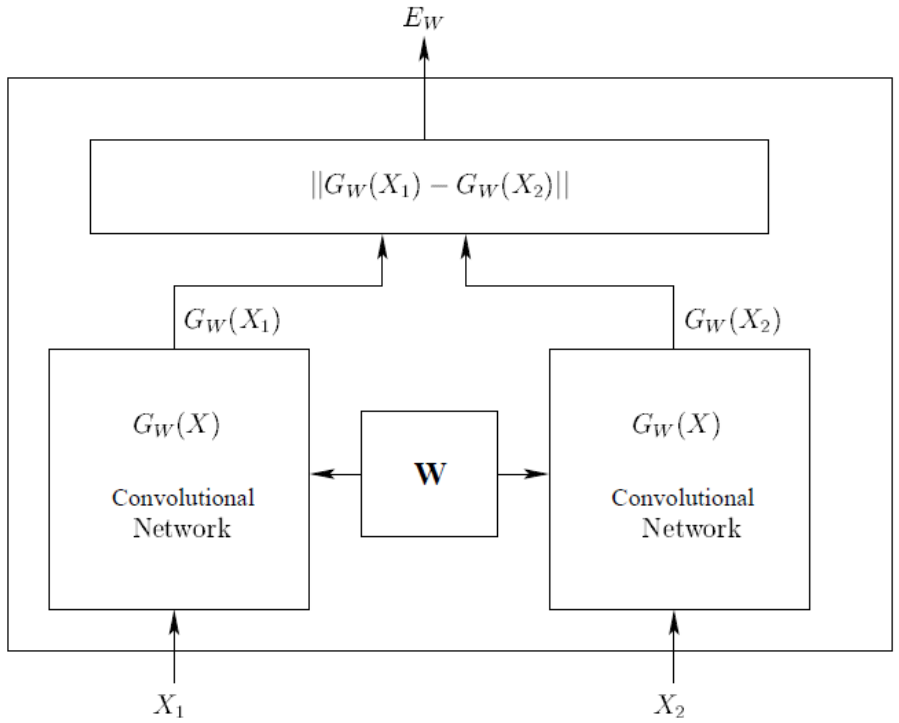


Raia Hadsell, Sumit Chopra, Yann LeCun, Dimensionality Reduction by Learning an Invariant Mapping. CVPR 2006

Siamese Architecture

- Given a family of functions $G_W(X)$ parameterized by W, find W such that the similarity metric $D_W(X_1, X_2)$ is small for similar pairs and large for dissimilar pairs:-

$$D_W(X_1, X_2) = \|G_W(X_1) - G_W(X_2)\|$$



Loss function

$$\mathcal{L}(W) = \sum_{i=1}^{p} L(W, (Y, \overrightarrow{X_1}, \overrightarrow{X_2})^i)$$

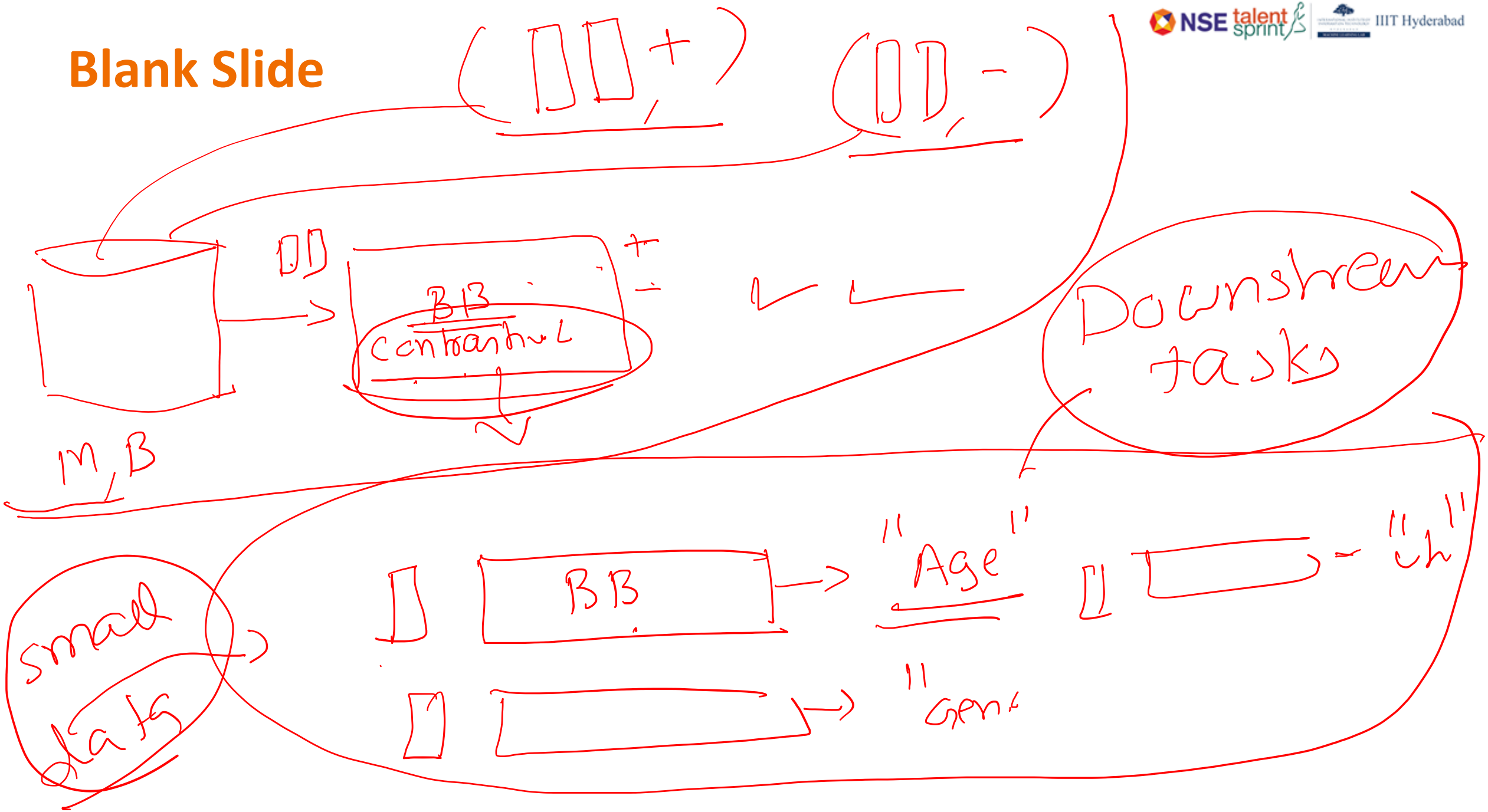
Loss function for similar pairs

$$L\left(W, (Y, \overrightarrow{X_1}, \overrightarrow{X_2})^i\right) = (1 - Y)L_s(D_W^i) + YL_D(D_W^j)$$

Loss function for dissimilar pairs

Raia Hadsell, Sumit Chopra, Yann LeCun, Dimensionality Reduction by Learning an Invariant Mapping. CVPR 2006

Blank Slide

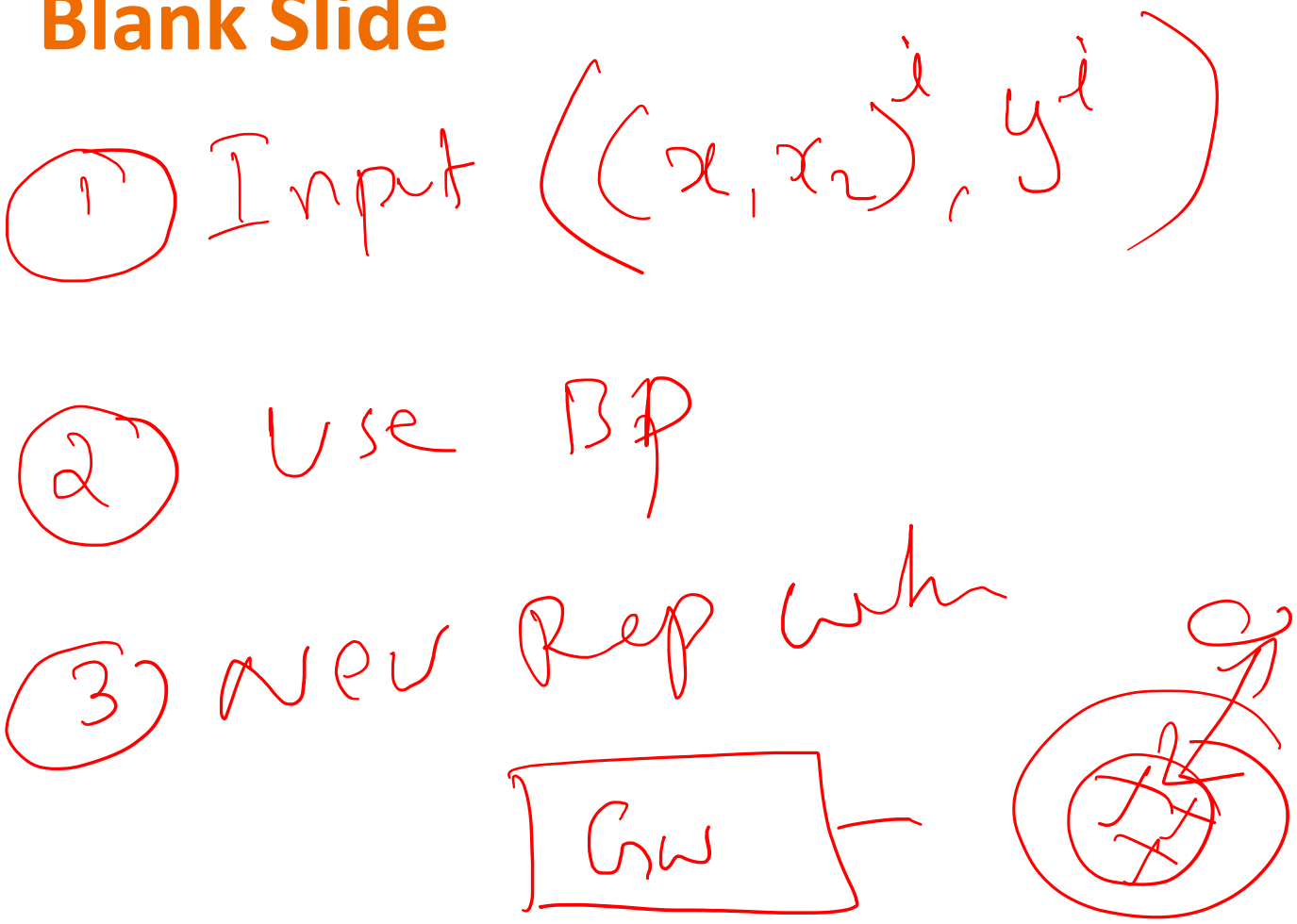


Contrastive Loss function

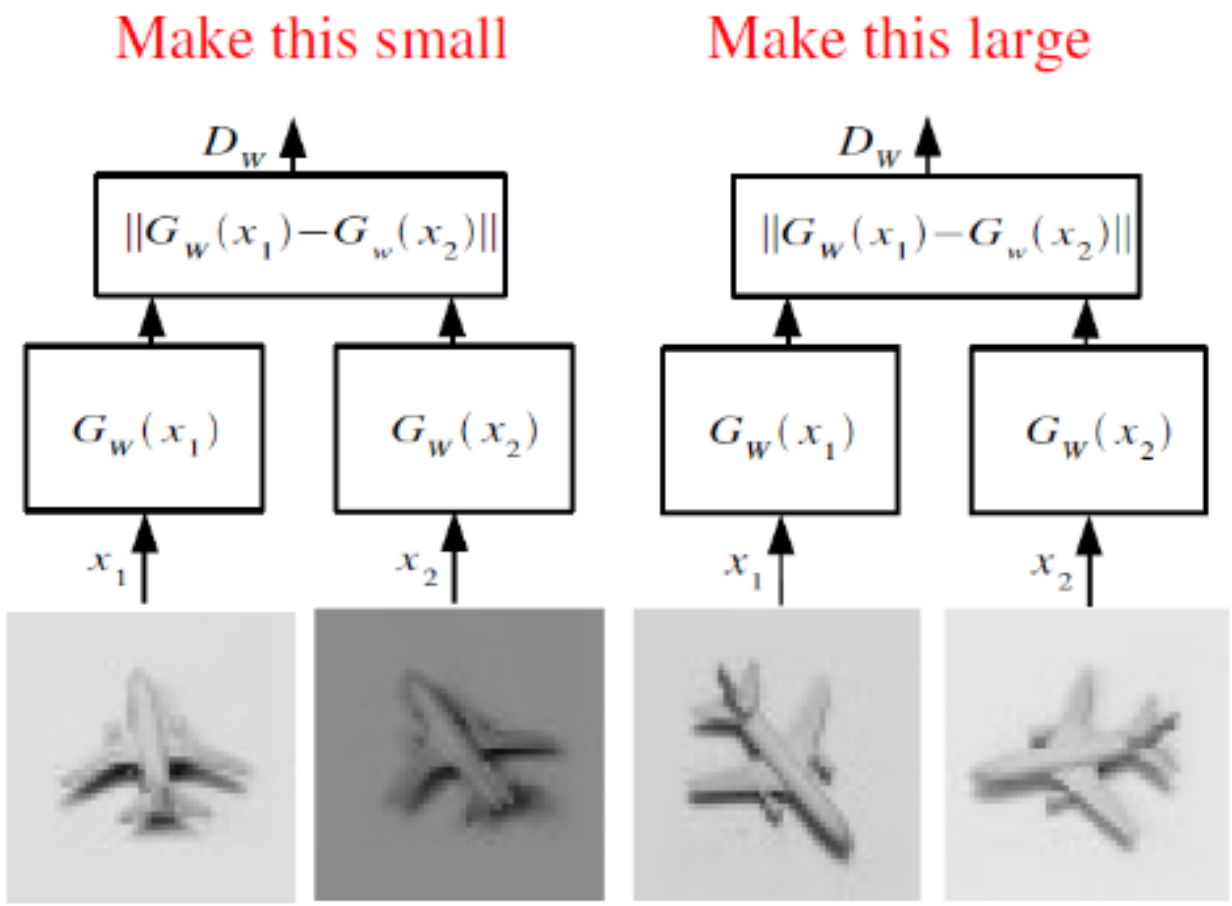
- Given $D_W(X_1, X_2)$ is the distance between the pair of samples, the contrastive loss function is defined as follows: (Y=0 for Similar and Y=1 for dissimilar pairs)

$$(1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{\max(0, m - D_W)\}^2$$

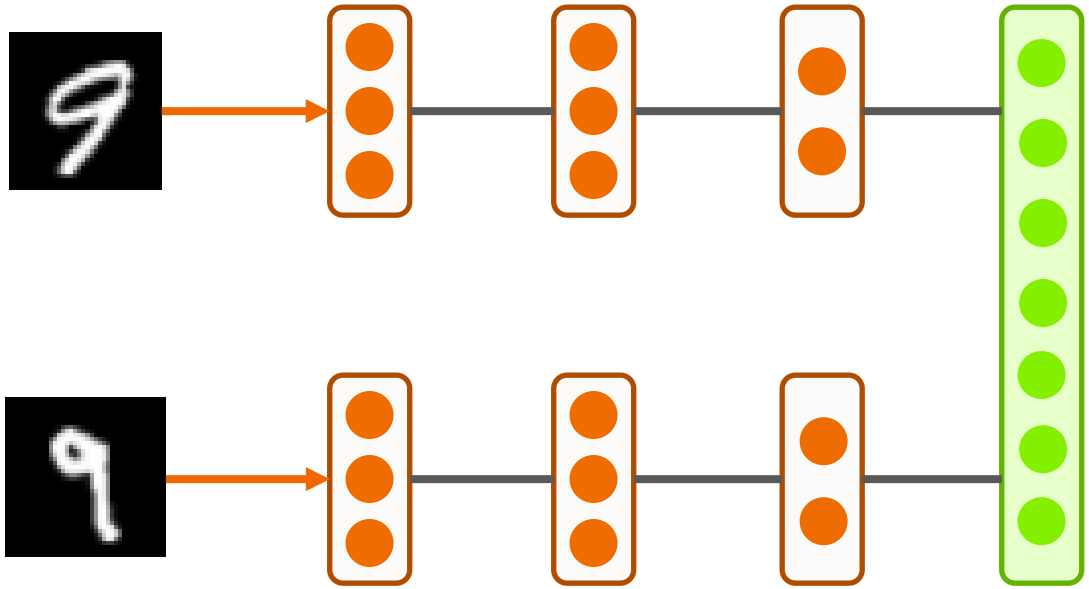
Blank Slide



Siamese: Loss



Example: MNIST

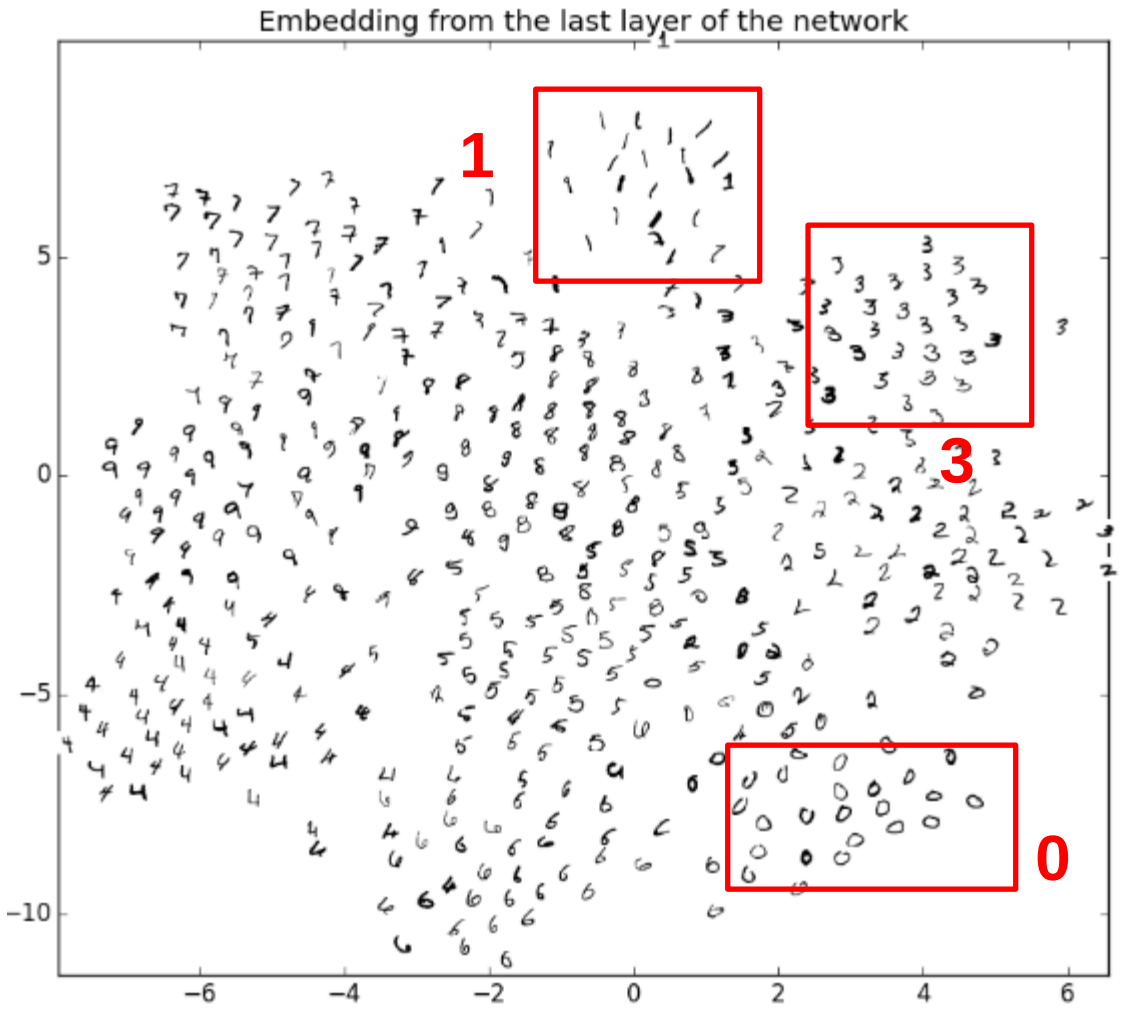


FC1 (1024 units)

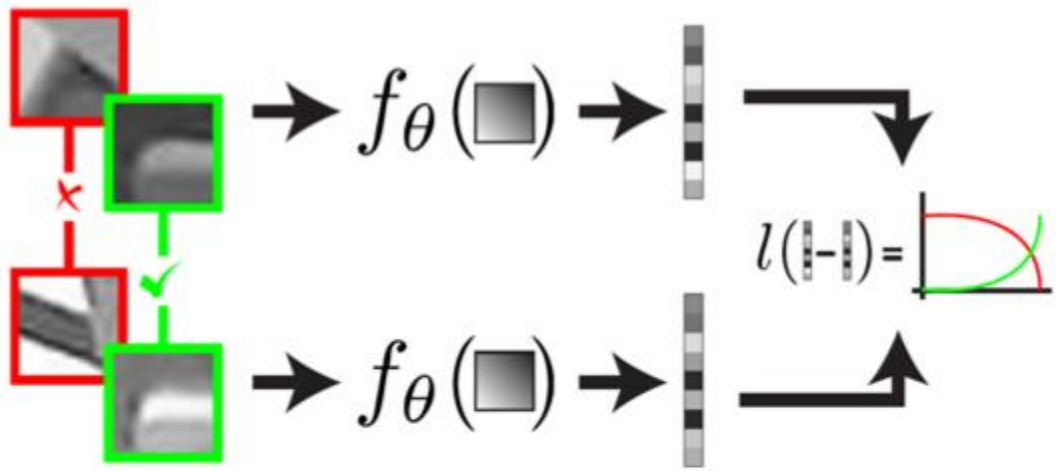
FC2 (1024 units)

FC3 (2 units)

Loss (contrastive loss)

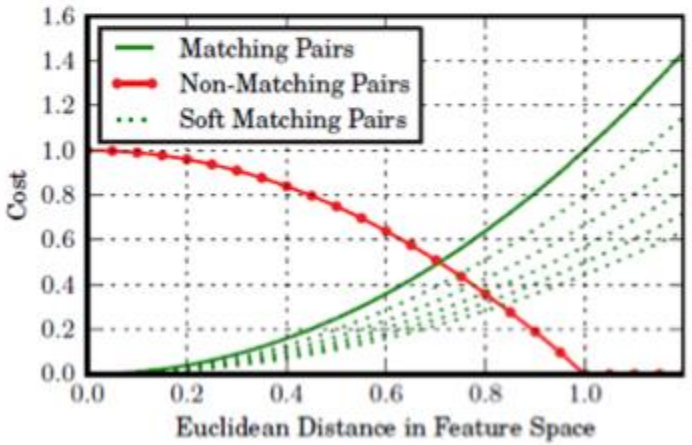


Application: Learning to Match Siamese Network



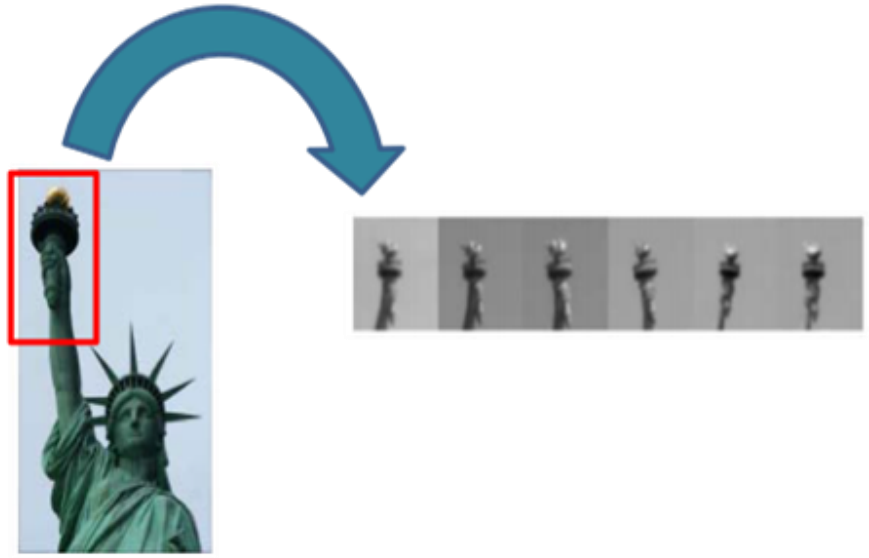
Using the contrastive cost function

$$l_{\boldsymbol{\theta}}\left(\mathbf{y}_i, \mathbf{y}_j\right) = \begin{cases} s_{ij} d_{ij}^2, & \text{if matching} \\ \max\left(1.0 - d_{ij}^2,\ 0\right), & \text{if non-matching} \end{cases}$$



IROS 2014

Matching across views



Learn a discriminative representation of patches from different views of 3D points

Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P. and Moreno-Noguer, F., 2015. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 118-126).

Person identification

CUHK03 Data set

True positive

True negative

Ahmed, E., Jones, M. and Marks, T.K., 2015. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3908-3916).

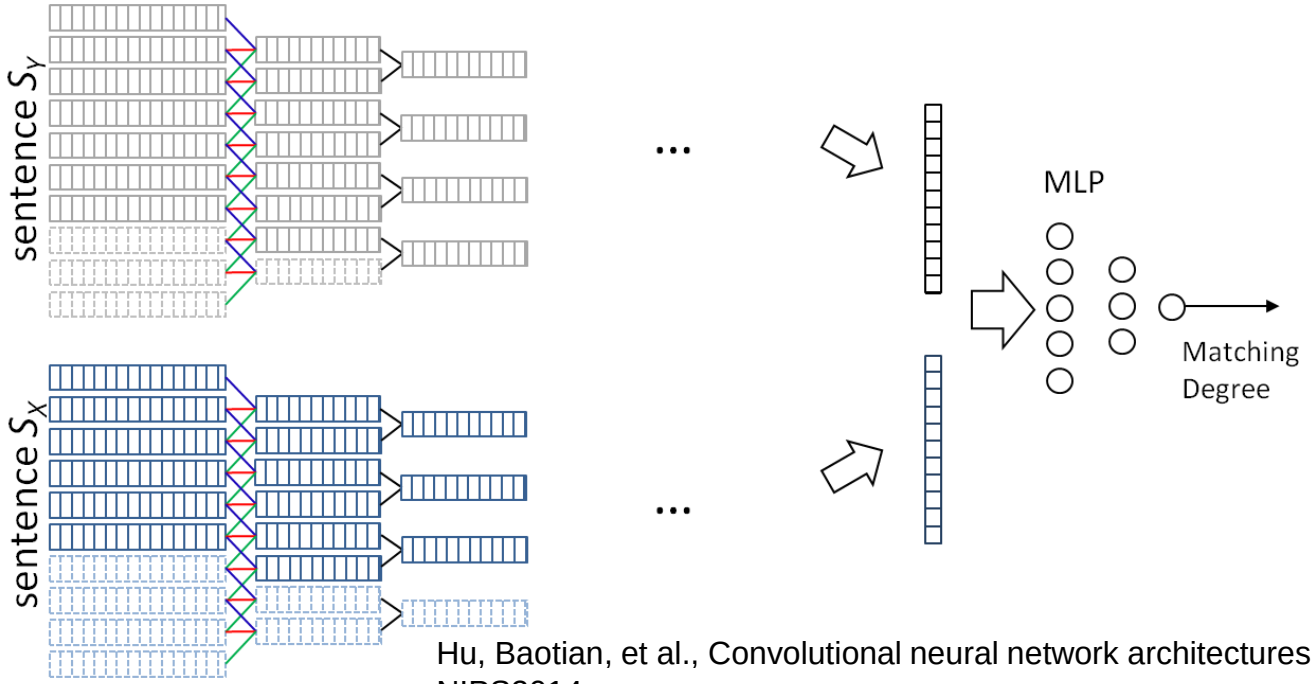
Application

- **Application:** Sentence completion, response to tweet, paraphrase identification

word2vec

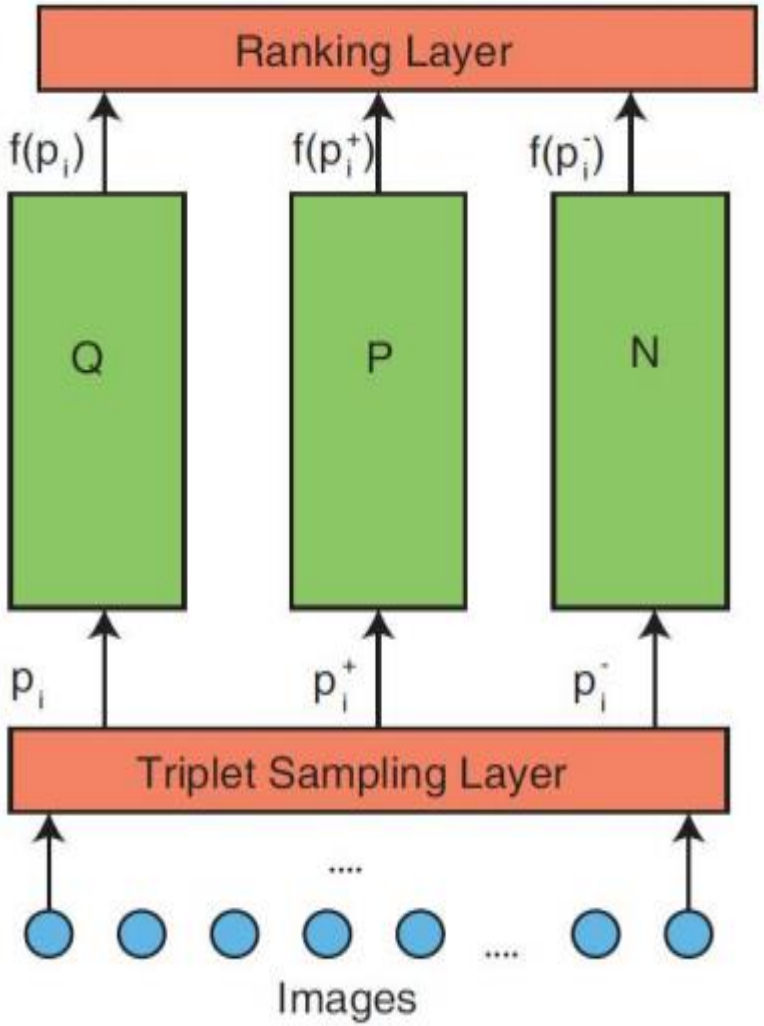
Example:

- x : Damn, I have to work overtime this weekend!
- Y^+: Try to have some rest buddy.
- y: It is hard to find a job, better start polishing your resume.



Hu, Baotian, et al., Convolutional neural network architectures for matching natural language sentences, NIPS2014

Triplet Loss



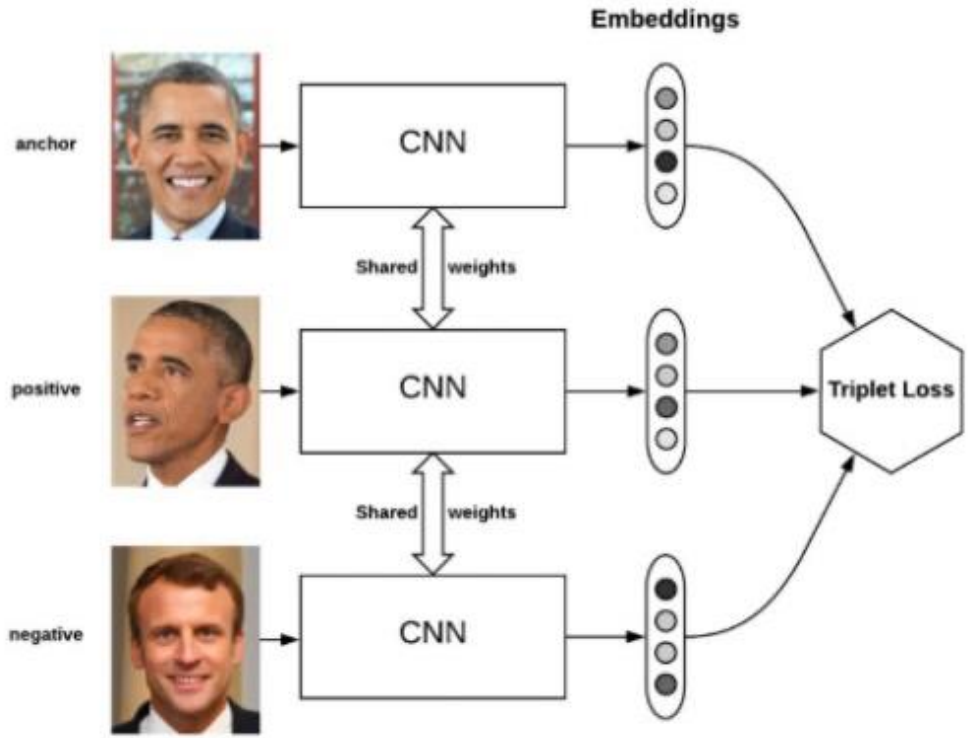
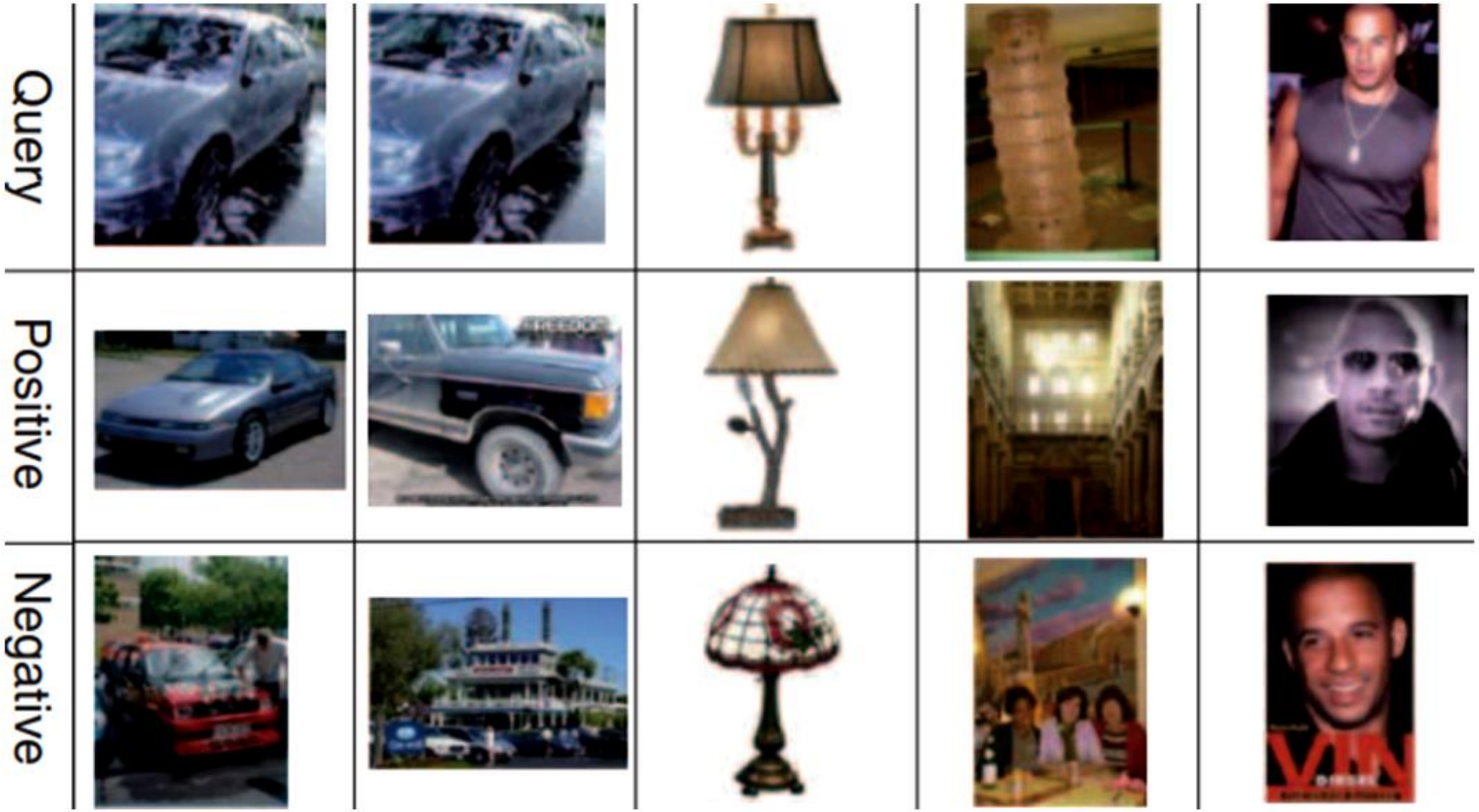


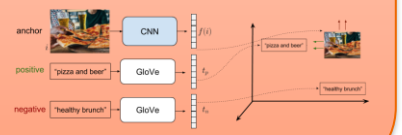
Image Retrieval (Ranking)



https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42945.pdf

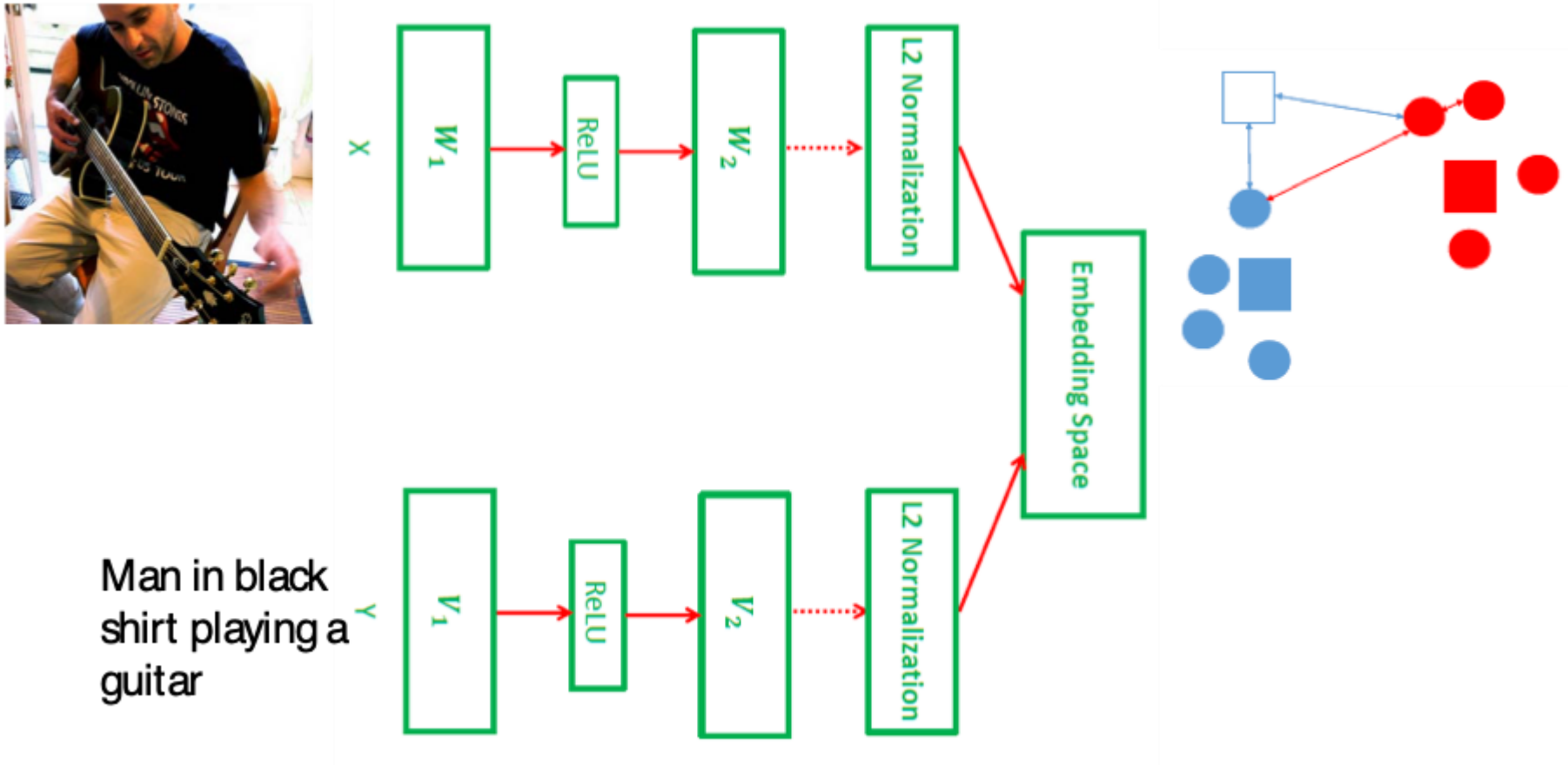
Acquire Raw Data → Prepare / Clean Data, Visualization → Feature Engineering → Pick Model and Hyper-params for Task → Model Training / Optimization → Evaluate Model Performance → Deploy Model

- ✓ Siamese Network
- ✓ Focus: Face
- ● Multimodal



Extension: Multimodal

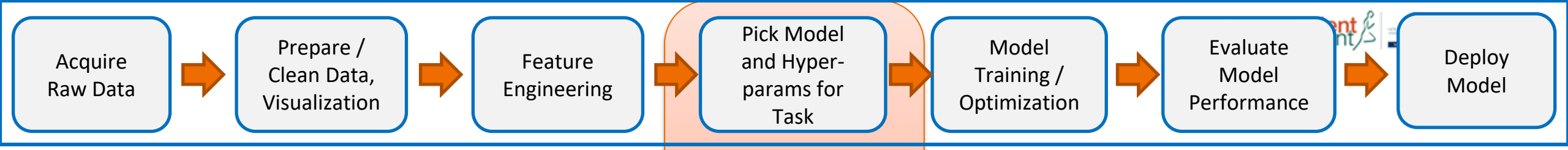
Cross modal matching



Wang, L., Li, Y. and Lazebnik, S., 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5005-5013).

Summary

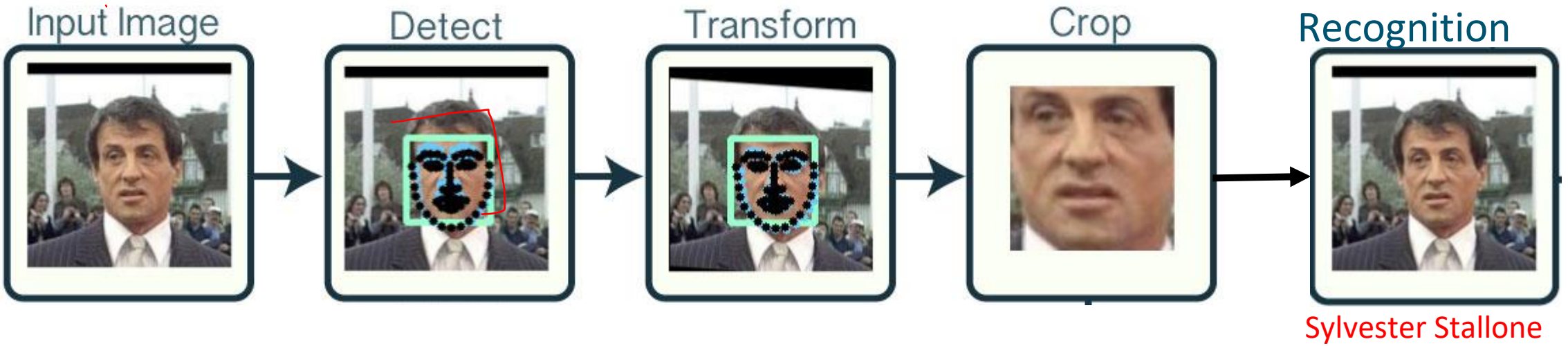
- Siamese and Triplet networks/losses are popular for solving fine grain classification (e.g. Face), capturing subjective needs (eg. Invariant to rotation), rankings etc.
- Available in many frameworks.
- Number of examples increase (e.g. ${}^{n}C_{2}$, ${}^{n}C_{3}$). Finding right pairs/triplets is also important.



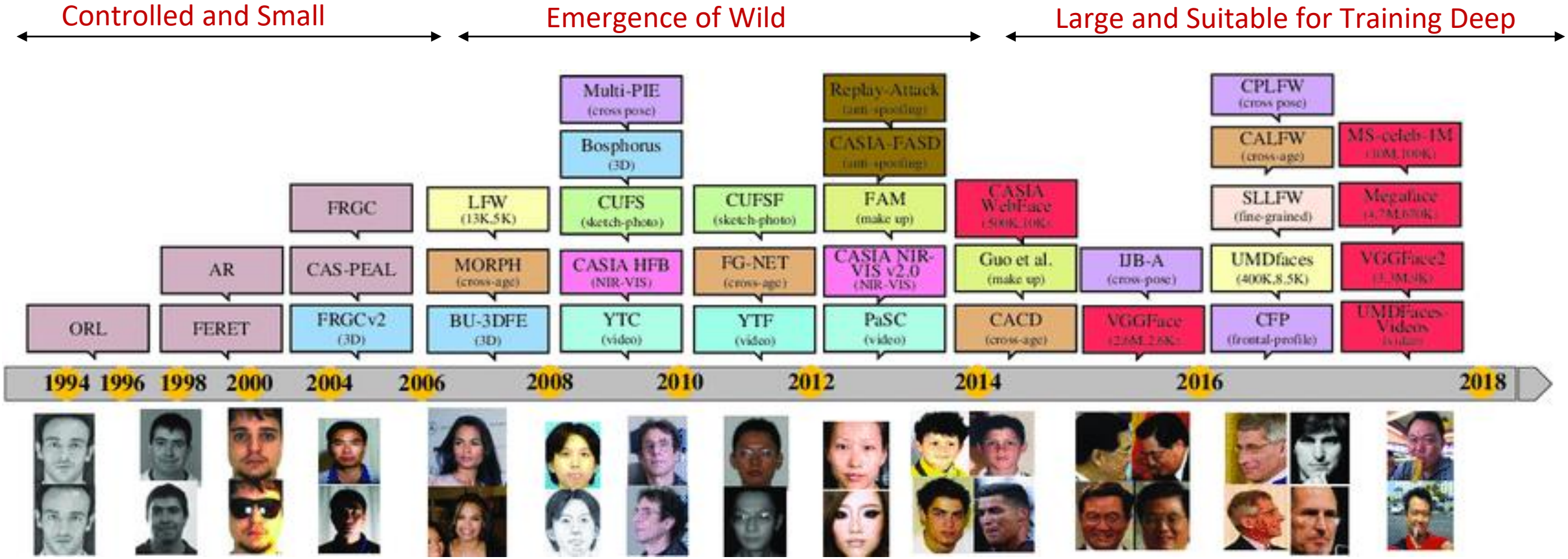
Example: Face

Focus on a specific area to appreciate the developments

Face processing pipeline

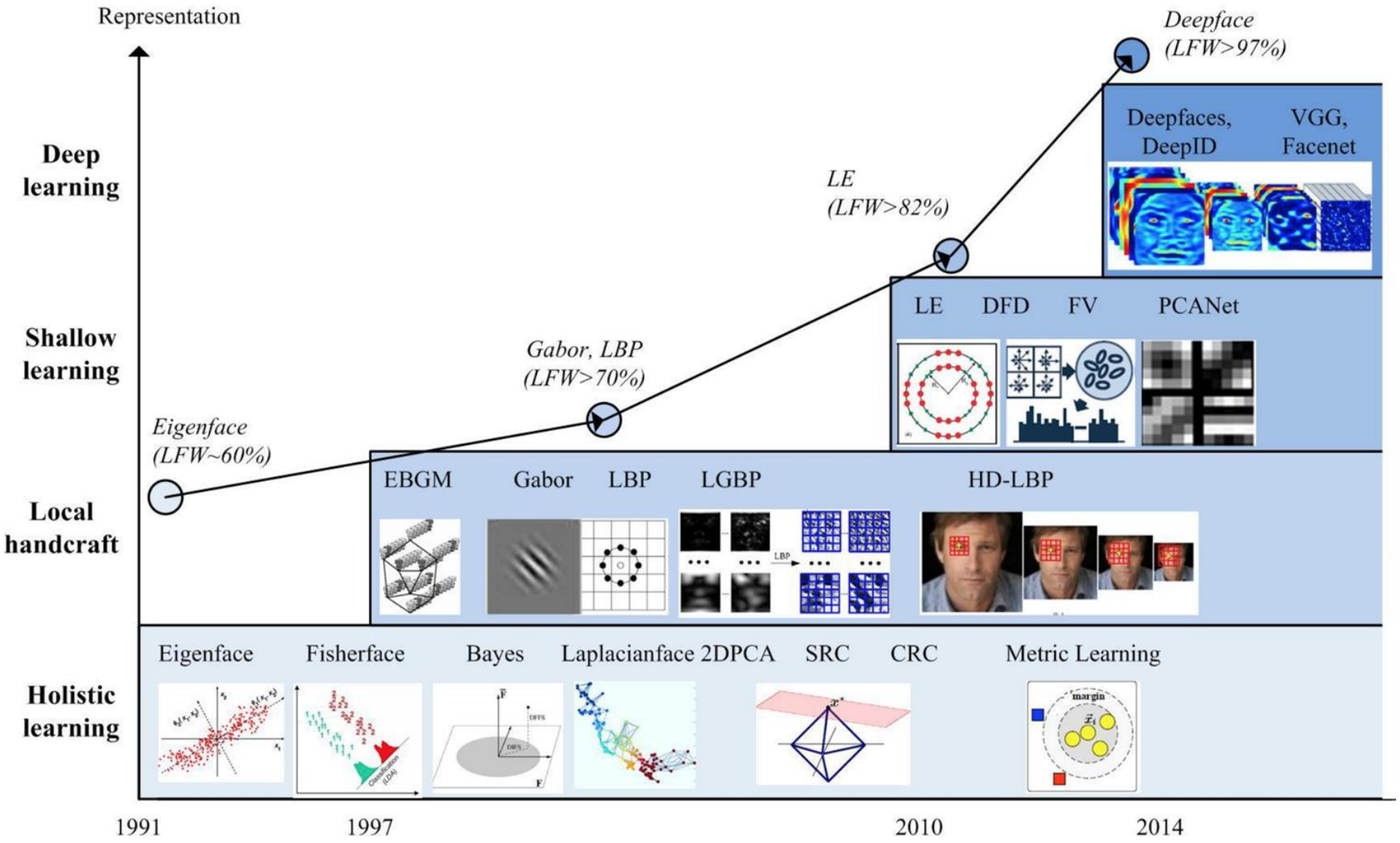


Evolution of FR Datasets



Wang et al. Deep face recognition a survey. arXiv 2018

Performance on LFW



Wang et al. Deep face recognition a survey. arXiv 2018

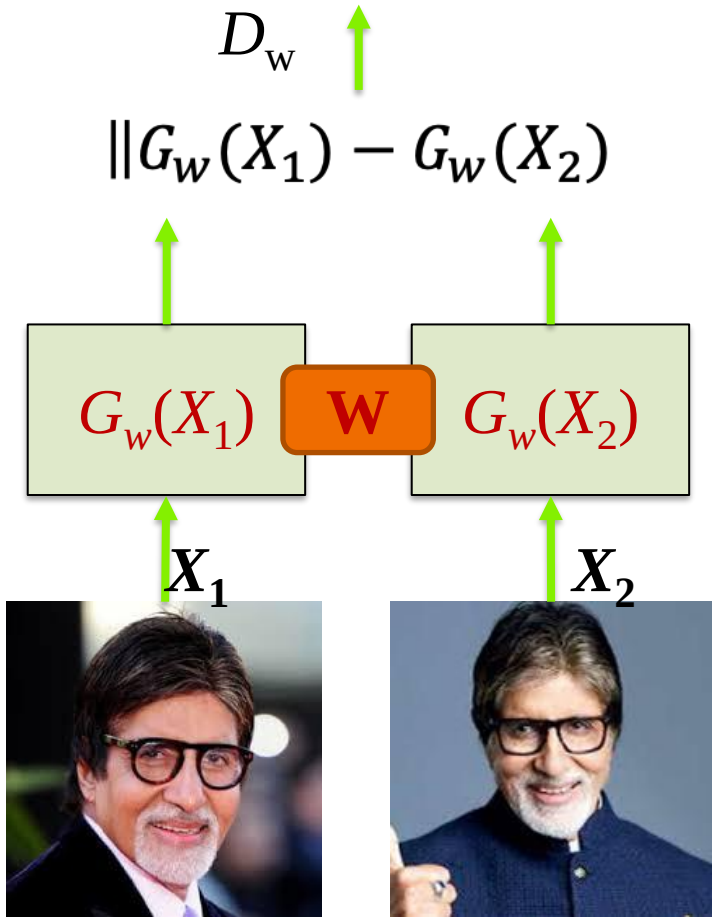
Systematic Evolution of Performance (LFW)

Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [20]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [21]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [36]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [38]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [58]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [37]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [85]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [101]	2016	center loss	Lenet+-7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [104]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [82]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [109]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [110]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [112]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [115]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [116]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [84]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [113]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [105]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [107]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [106]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [117]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50

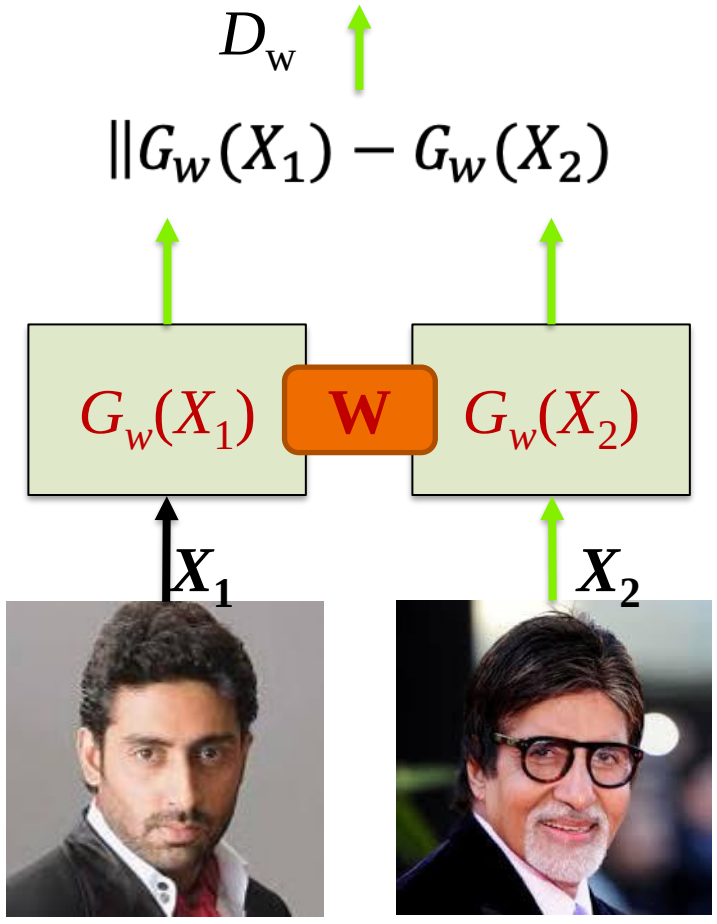
M. Wang and W. Deng, “Deep Face Recognition: A Survey”, Arxiv, 2018

Siamese Loss

Make this smaller

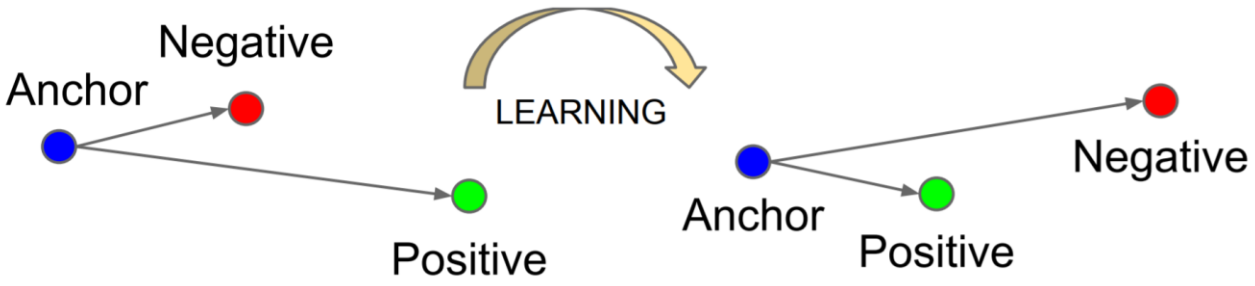


Make this larger



- Only pair-wise Labels
- Similarity Metric:
  $D_w(X_1, X_2)$
- Have shared weights
- Training in batches

Triplet loss function



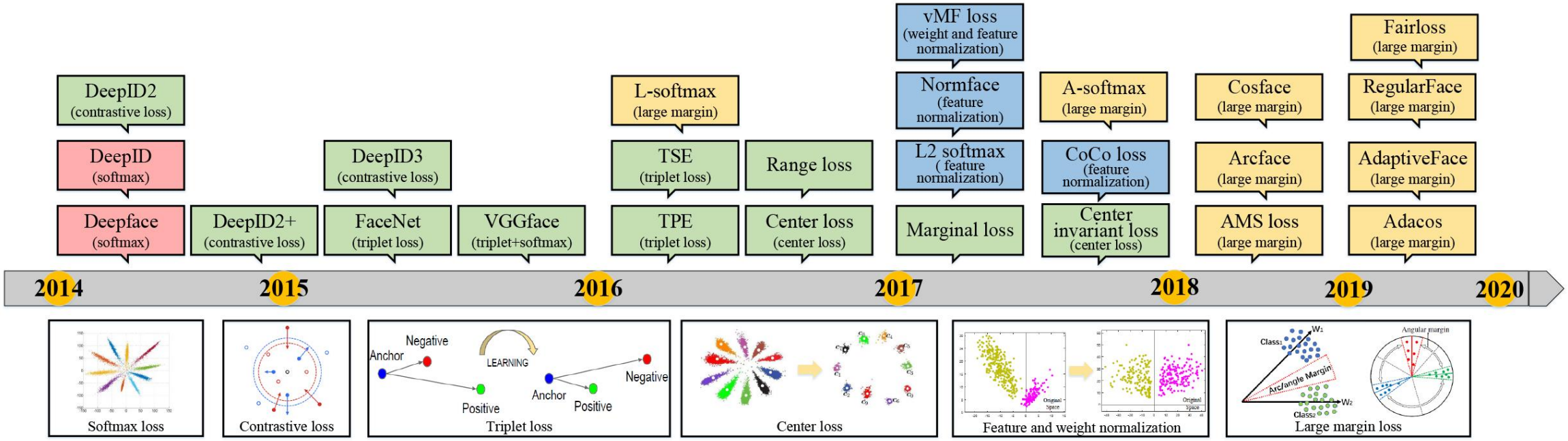
$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \ \forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T}$$

where *f* is the embedding

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

Florian Schroff et al. FaceNet: A Unified Embedding for Face Recognition and Clustering, CVPR 2015

Summary: Evolution of Loss Functions

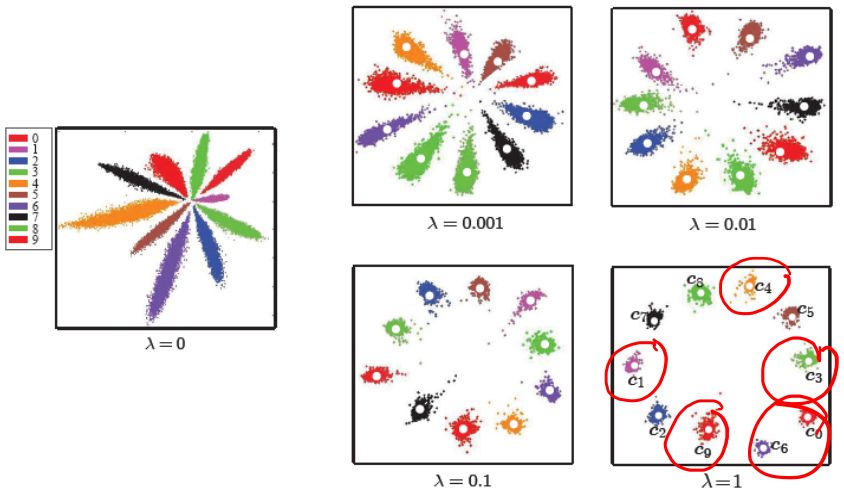


https://arxiv.org/pdf/1804.06655.pdf

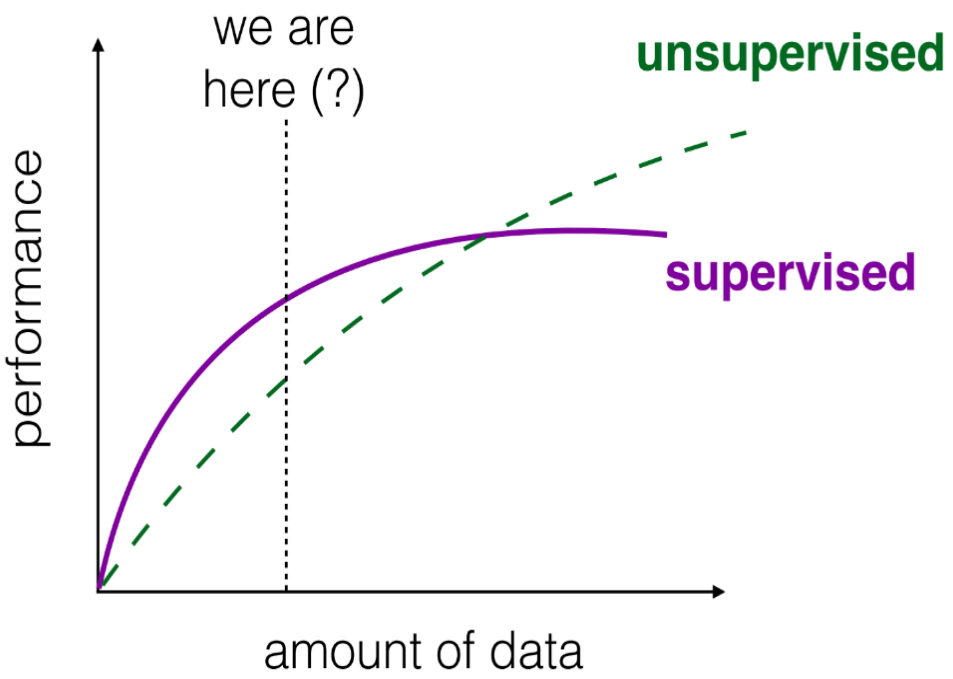
Center Loss

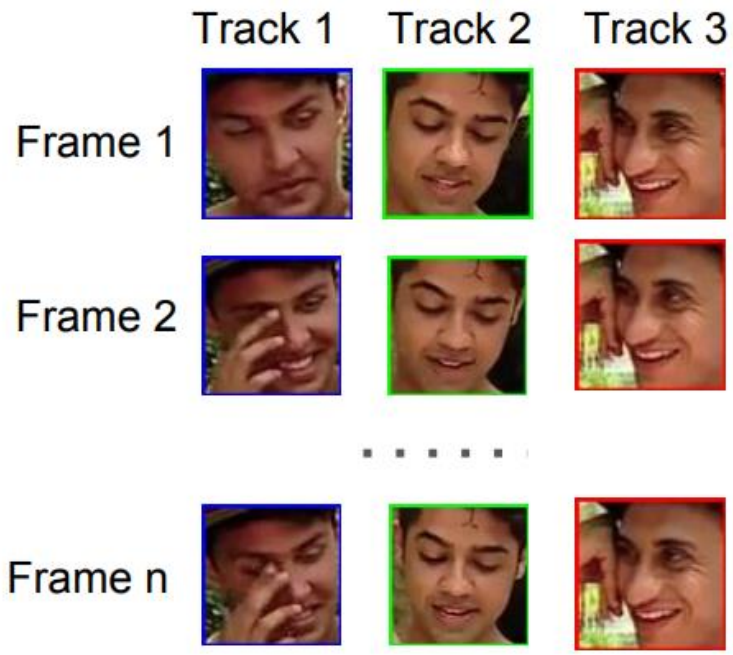
$$L = L_s + \mu L_c = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} ||x_i - c_{y_j}||_2^2$$

Two Terms (1) For Interclass separability (softmax) and (2) Intraclass compactness and Lambda that balance between them



Unsupervised Learning of Face Representations

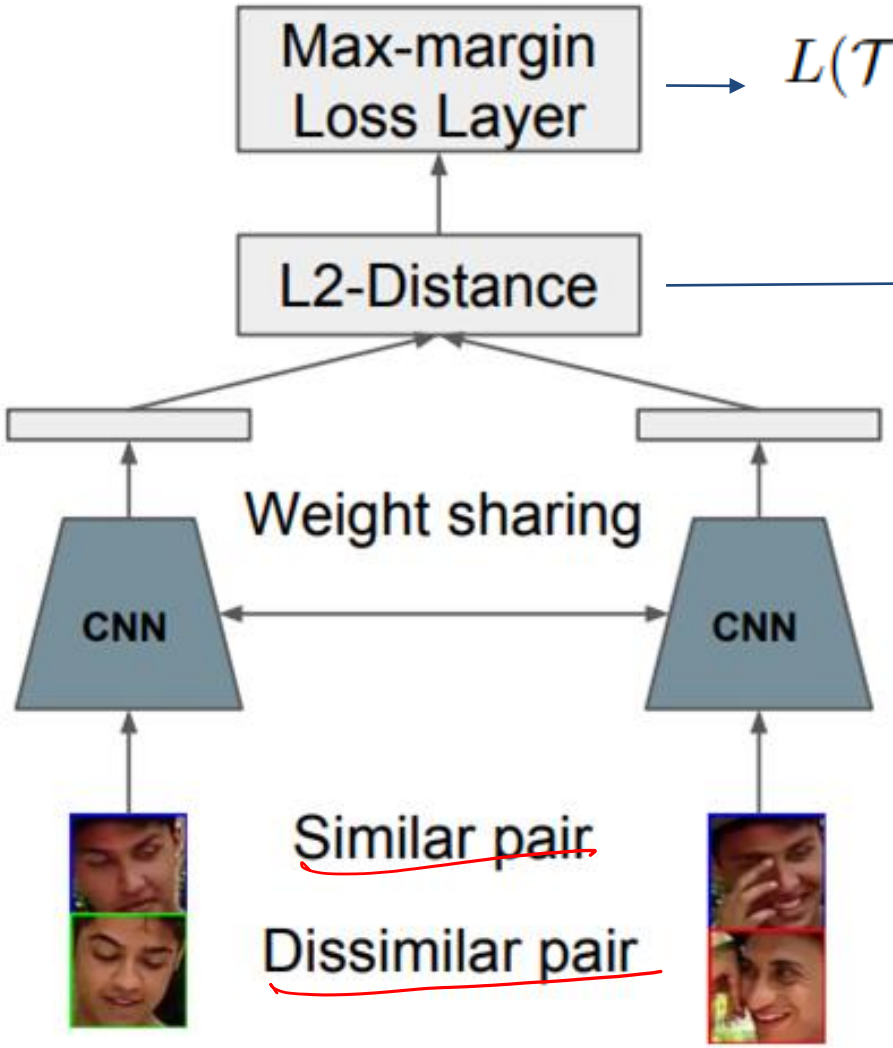




Learn from YouTube

1. Multiple faces in the same frame must belong to different people
2. The same face tracked across multiple frames belongs to the same identity

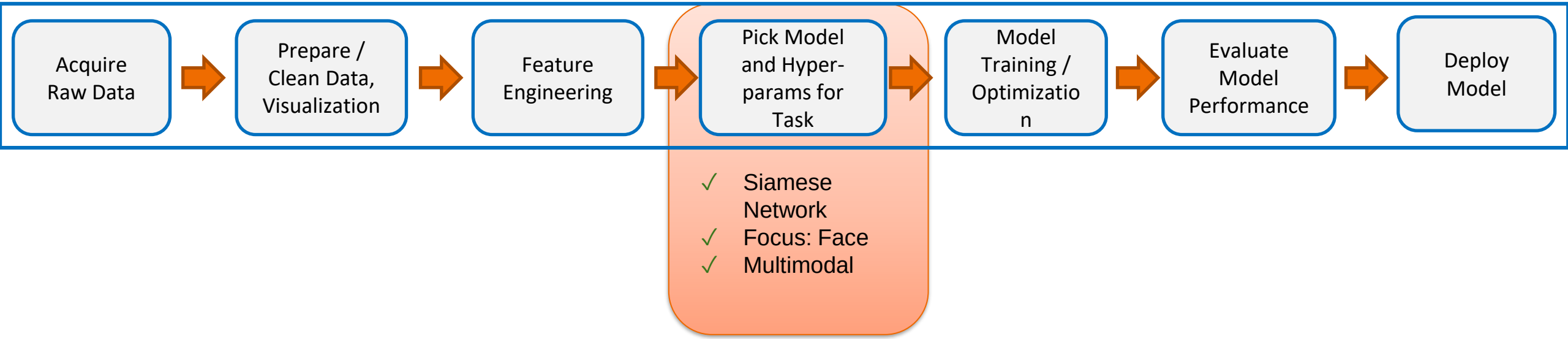
Model



$$L(\mathcal{T}) = \sum_{\mathcal{T}} \max\left\{0, m - y_{ij}(b - D^2(\mathbf{x}_i, \mathbf{x}_j))\right\}$$

$$D^2(\mathbf{x_1}, \mathbf{x_2}) = \|\phi(\mathbf{x_1}) - \phi(\mathbf{x_2})\|_2^2$$

Summary


Acquire Raw Data
Prepare / Clean Data, Visualization
Feature Engineering
Pick Model and Hyper-params for Task
Model Training / Optimizatio n
Evaluate Model Performance
Deploy Model
✓ Siamese Network
✓ Focus: Face
✓ Multimodal

Learning Under Practical Constraints

Recap: Learned Representations

Learned Representations

CNN Features can be used for wider applications:

- Train the CNN (deep network) on a very large database such as imageNet.

- Re-use CNN to solve smaller problems
  - Remove the last layer (classification layer)
  - Output is the code/feature representation

New Settings

- Extend to more classes
  - Extend from 1000 classes (say people) to another new 100
- Extend to new tasks
  - Extend from object classification to scene classification
- Extend to new data sets
  - Extend from imageNet to PASCAL (SLR to webcams)
- When we have a lesser amount of data.

Transfer learning

- Same domain, different task
- Pre-trained Image Net (visual domain of real images)
  - Train on image classification
- Extend to Classification on a new category
  - Q: How much can be reused?
- Fine-tune on new task
  - E.g., semantic image segmentation
  - > keep ‘backbone the same, fine-tune ‘head’ layers
  - > assumption: visual features generalize within domain

Fine Tunining: Challenges

- Fine-tune on new task
  - Often need to train entire network, because input features will be different
  - Training only a few layers at the end is less likely to fundamentally solve it
- How much labelled data in the target domain?
  - Zero-shot learning
  - One-shot learning
  - Few-shot learning

Examples of Practical Issues

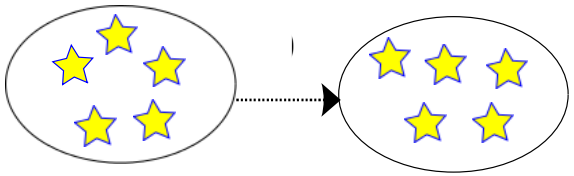
- Amount of Data available for Learning
  - Very small amount is available in many situations
  - Few shot learning; Zero shot learning, One shot learning
- Limited Supervision
  - Amount of supervision (small annotated and more unannotated)
  - Cost of annotation (how do we minimize the cost; human in the middle)
  - Type of annotation (Eg. weak annotation and noisy annotation)
- Newer Situations
  - Training was done in a different dataset
  - Eg. Domain adaptation, Transfer learning

Why?

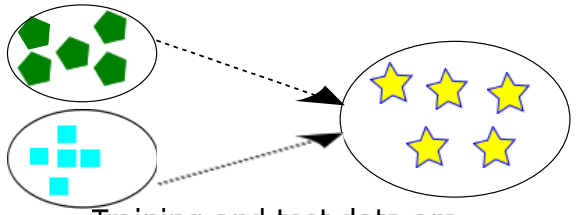
An example

- We have **labeled** images from a **Web image** corpus
- Is there a `Person` in **unlabeled** images from a **Video** corpus ?

`Person` `no Person` Is there a `Person`?



Training and test data are from the same domain



Training and test data are from different domains

Challenge to Create in all situations

Proprietary

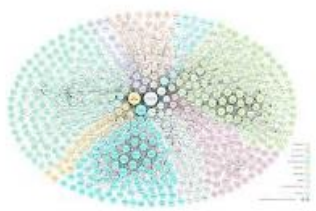
Private

Annotate Everything is “Impractical”

Active Learning

Passive Machine Learning

Raw, unlabeled data



Unlabeled data passed to oracle

Oracle

Labelled Data

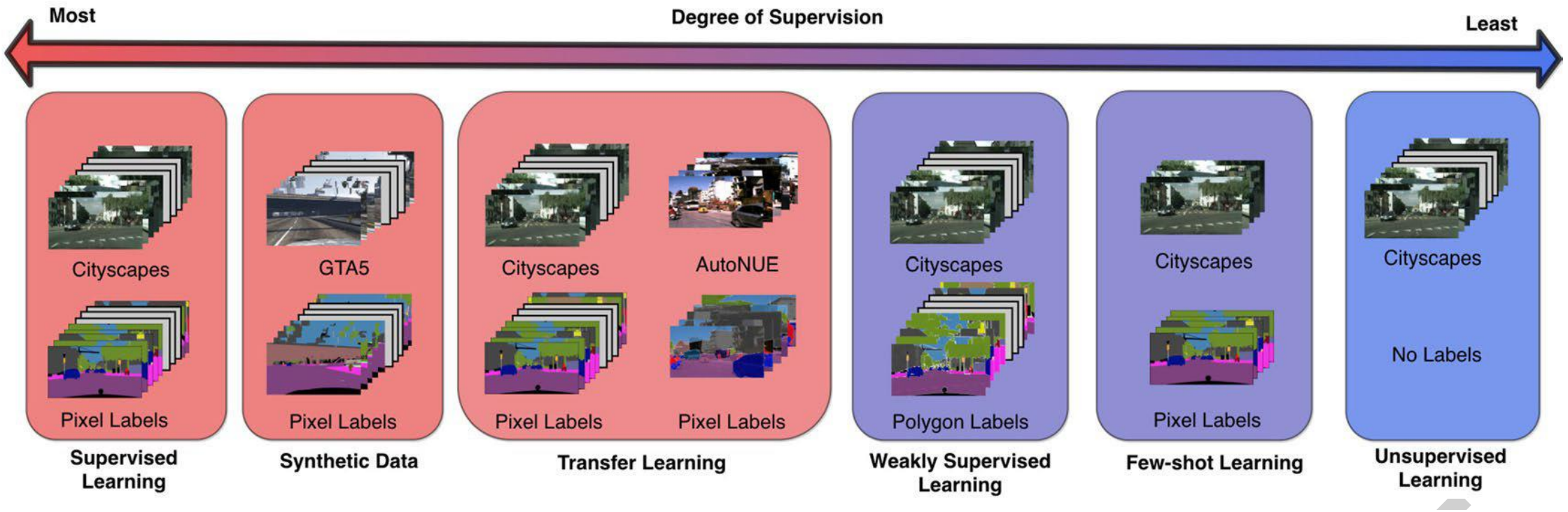
Machine Learning Model

Trained Classifier

Example task: Semantic Segmentation

Expensive task and challenging to have millions of examples in all situations

Space of Supervision



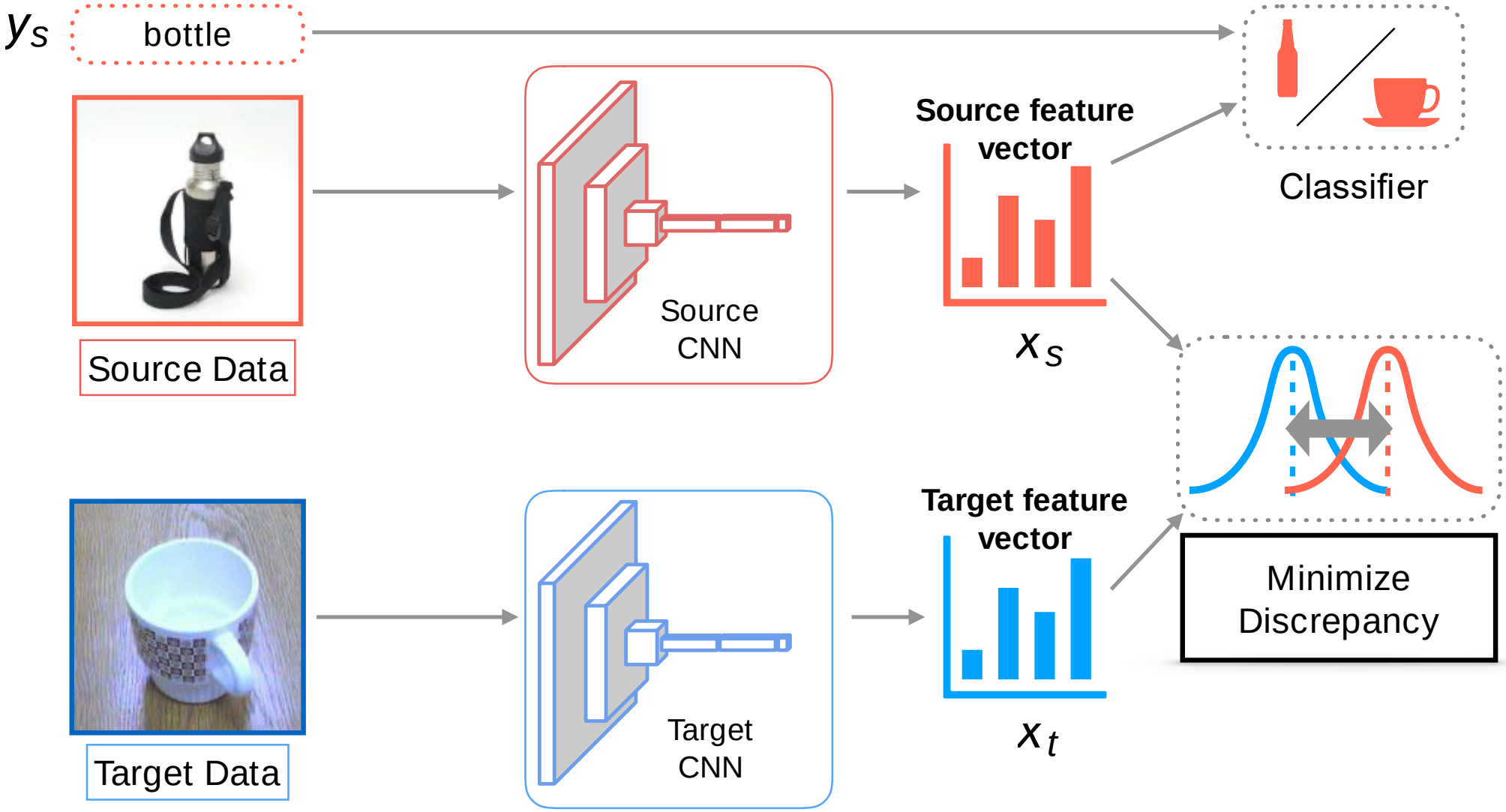
There is a need to adapt

Adapt

Source Domain: Lots of Training Data

Target Domain: Unlabeled or Limited Training Data

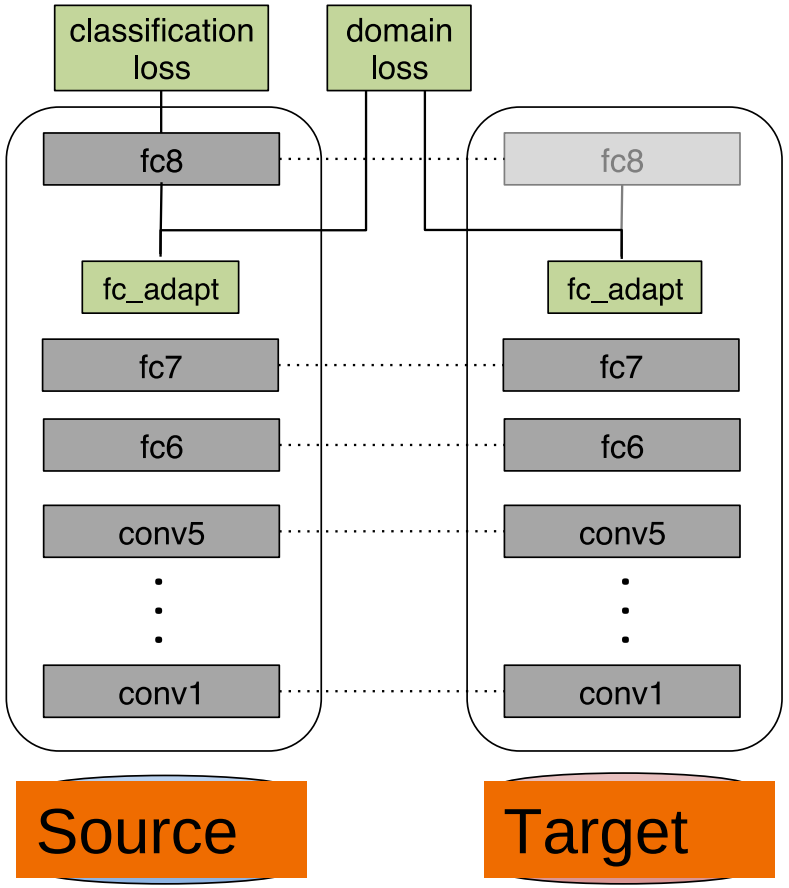
Deep Domain Adaptation



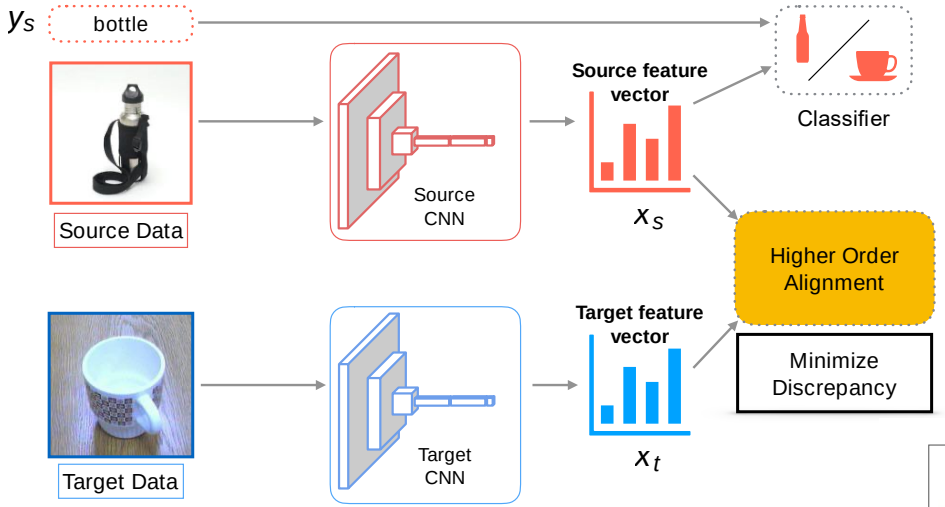
Maximum Mean Discrepancy (MMD)

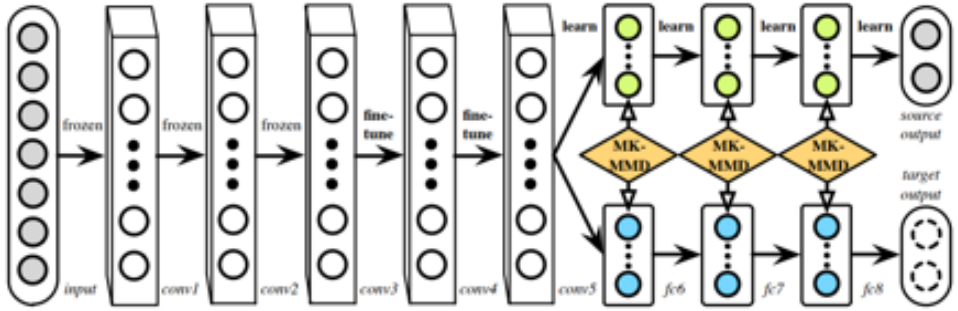
$$\text{MMD}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|$$

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \text{MMD}^2(X_S, X_T)$$

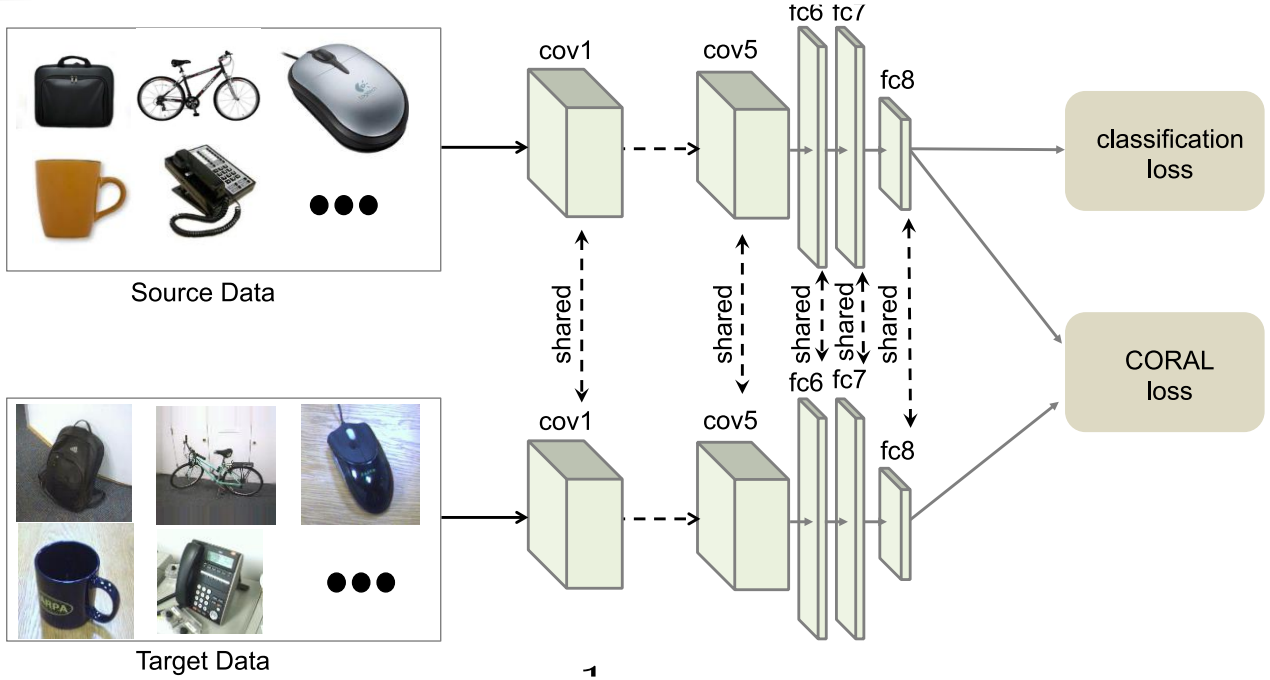


Aligning Higher Order Statistics





Kernelized MMD



Summary

- There are elegant and algorithmic results and methods available in many practical situation of our interest
  - Eg. Domain adaptation with no source data
- How do we approach in new situation?
  - Characterize the new situation properly
  - Learn to characterize the situation (than learn to solve the problem first)
- Often an effective strategy could be available already or directions for solving could be known well.

Thanks!!

Questions?