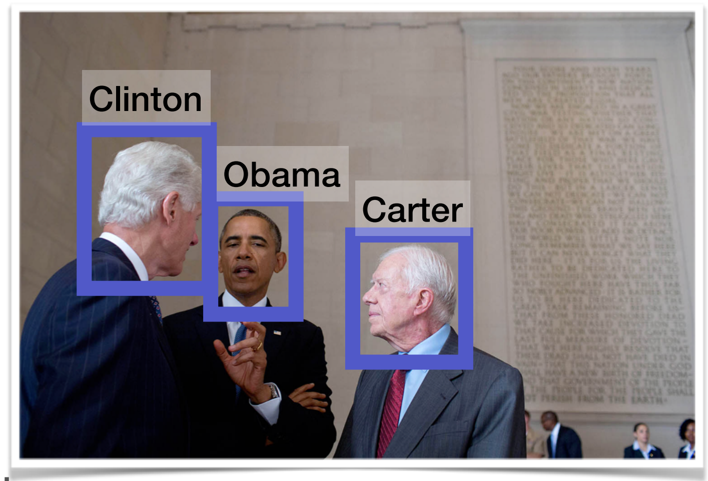
Similarity Learning with CNN’s

Example: Face classification

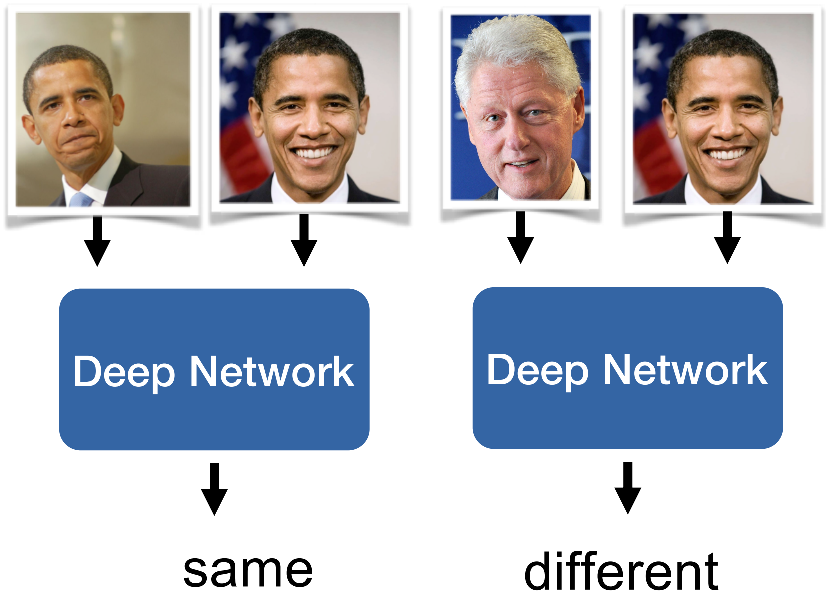
- Classify who is in a picture
  - Each person is a class

Bush Kennedy Washington



Issues

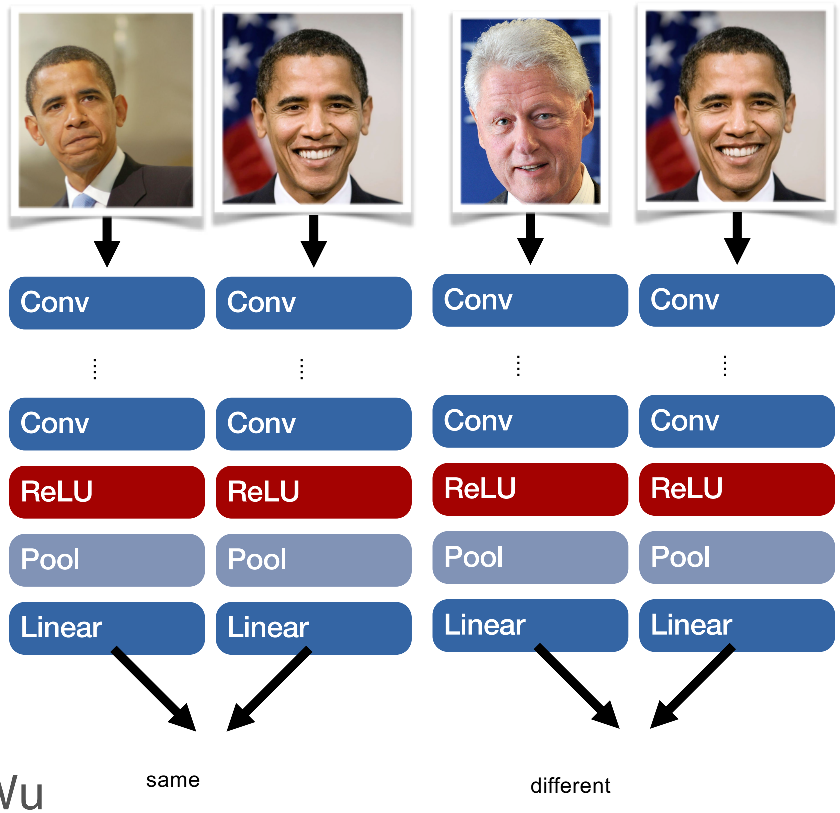
- What do we do when we have a new class?
  - Classifier needs to retrain
  - Instead try to learn similarity – use



Solution - Siamese networks

- Instead of having single network
- Separate network for each image a share the final layers
- Distance metric, cos, dot product
  - or KNN search

Signature Verification using a Siamese Time Delay Neural Network, Bromley et al., NIPS 1994

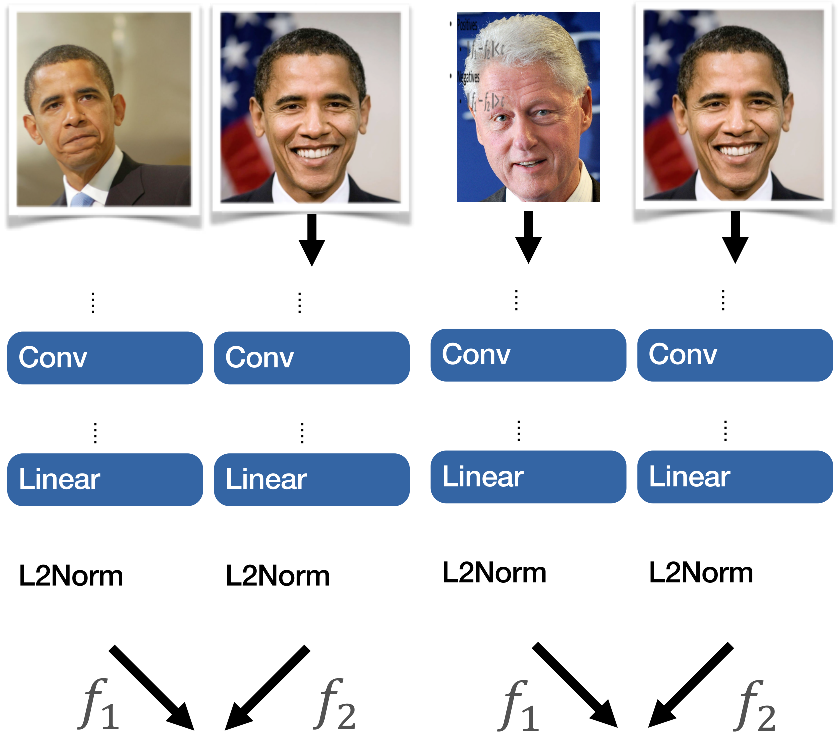


Objective, Contrastive Loss

- Positives
  - $\| f_1 - f_2 \| < c$
- Negatives
  - $\| f_1 - f_2 \| > c$

Contrastive Loss

- Collapse positives
  - $\| f_i - f_j \|$
- Separate negatives
  - $max(c - \| f_i - f_j \|, 0)$

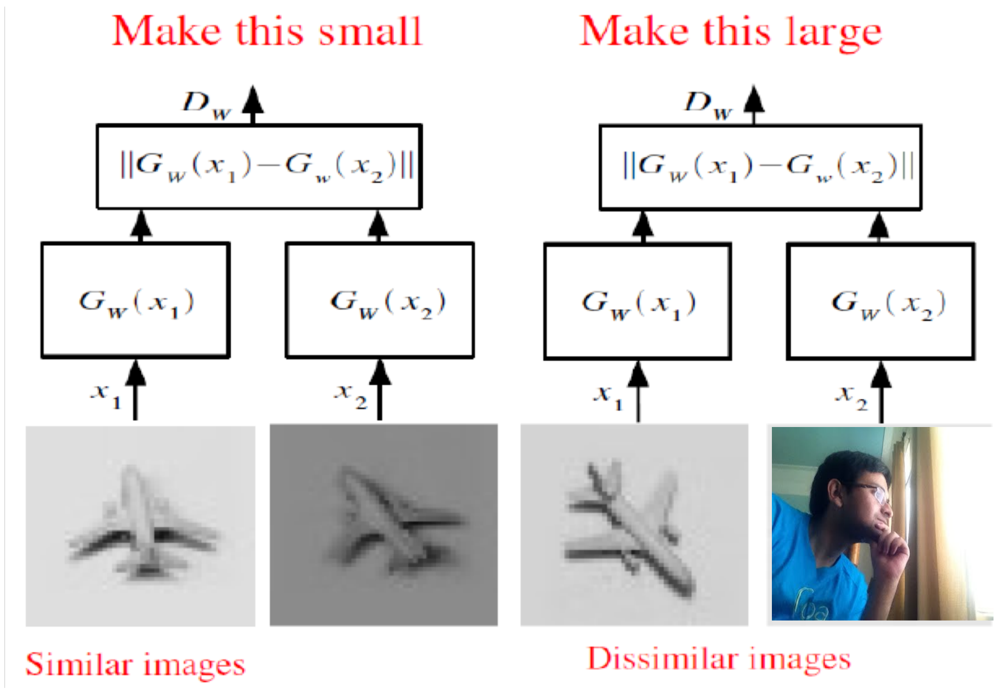


$y = \{0, 1\}$

$$L(x_p, x_q, y) = (1 - y).\max(0, m^2 - \|x_p - x_q\|^2) + y\|x_p - x_q\|^2$$

close far

Dimensionality reduction by learning an invariant mapping, Hadsell et al., CVPR 2006



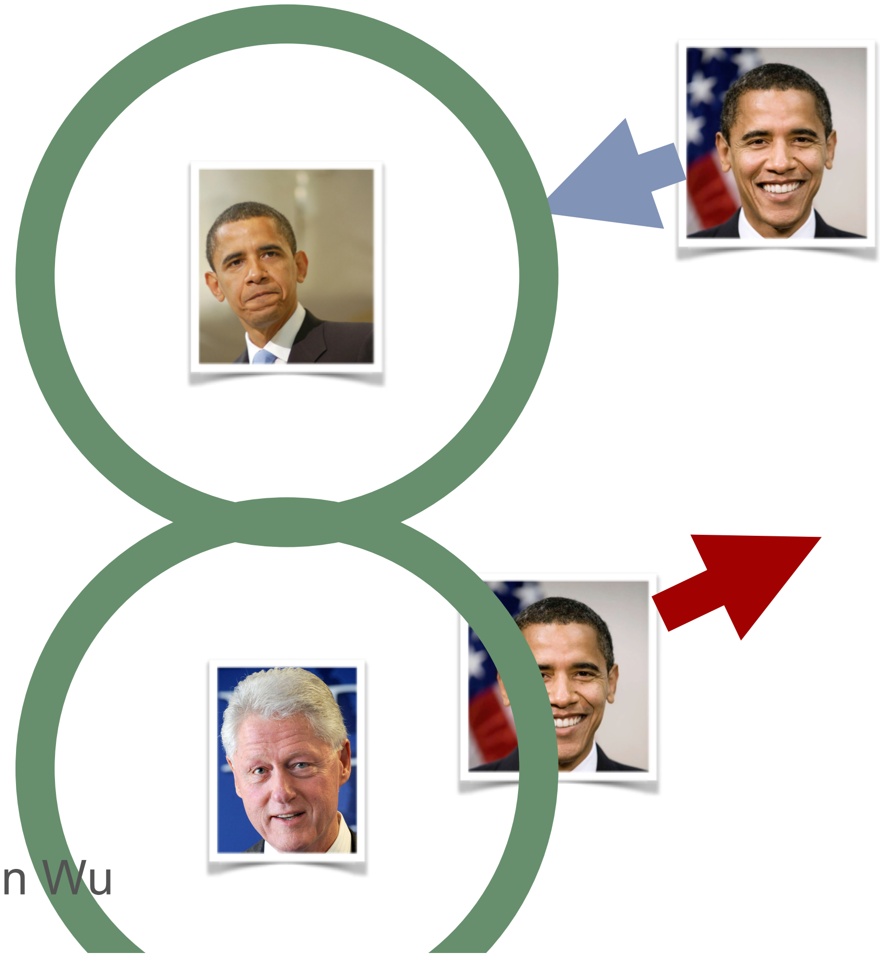
Chopra, S., Hadsell, R. and LeCun, Y., 2005, June. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 539-546). IEEE.

Margin based loss

- Collapse positives
  - $max(\| f_i - f_j \| - c, 0)$
- Separate negatives
  - $max(c - \| f_i - f_j \|, 0)$

Sampling Matters in Deep Embedding Learning, Wu et al., ICCV 2017

Problem: need to fix thresholds



Embedding learning, Triplet Loss

- Distances
  - Absolute distances don't matter at inference
  - KNN cares about relative distance
- Objective

$$max(0, \| f_i - f_j \| - \| f_j - f_k \| + \alpha)$$

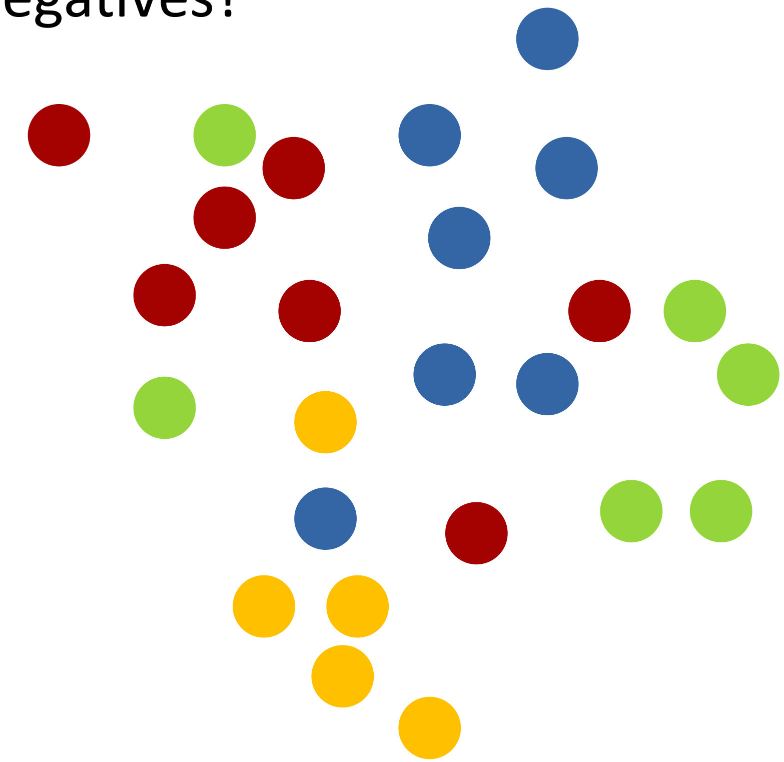
- Positive pair i, j
- Negative pair i, k

- Harder to train – need to consider all triplets
- Distorts the embedding space

- *Learning a distance metric from relative comparisons, Schultz and Joachims, NIPS 2003*
- *Distance metric learning for large margin nearest neighbor classification, Weinberger and Saul, JMLR 2009*

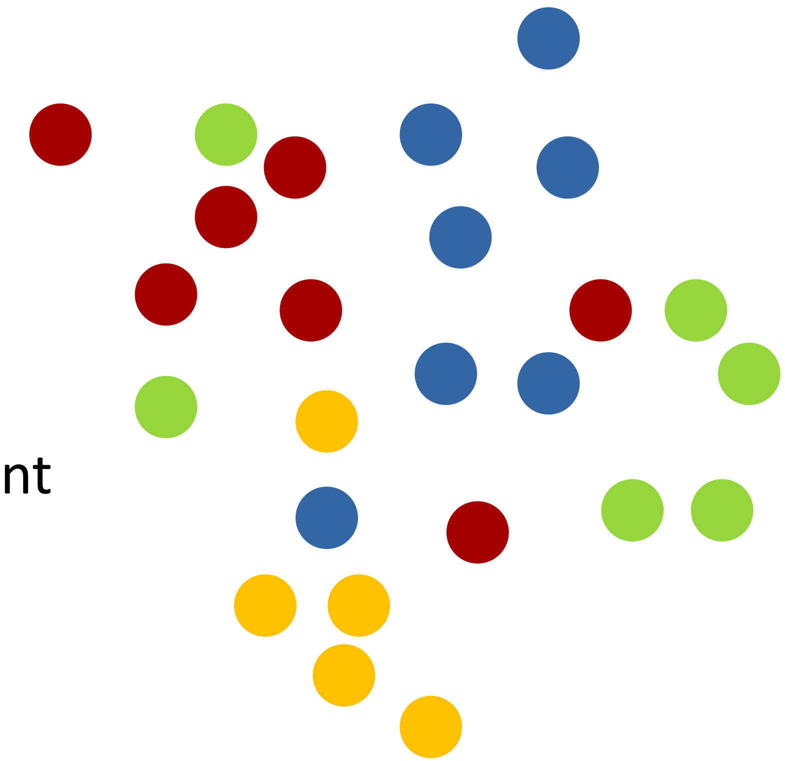
Sampling

- How do we select positives and negatives?
- All pairs, all triplets =, Bad idea
- very slow
    - Pairs $O(N^2)$
    - Triples $O(N^3)$
- Use sampling



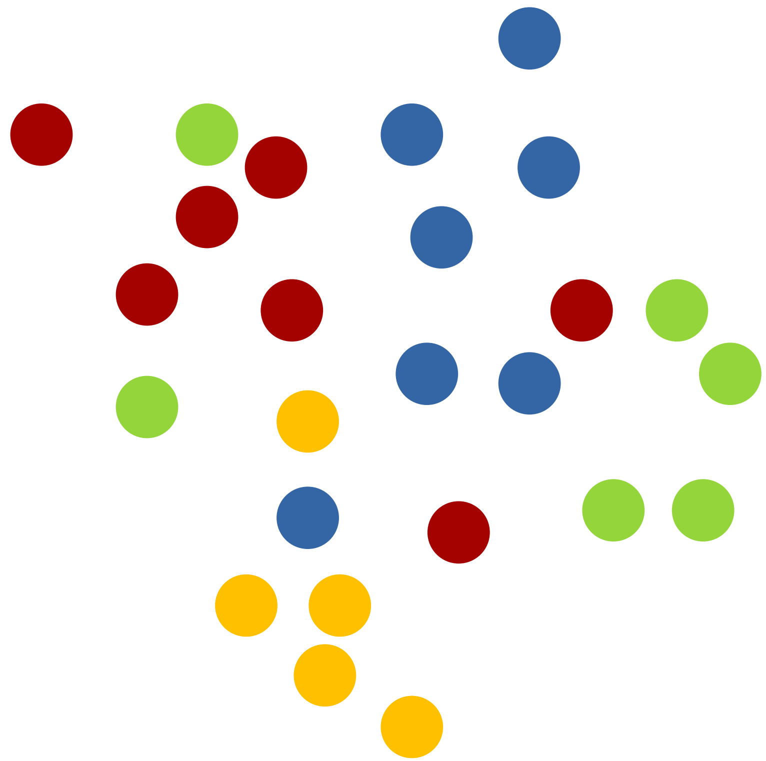
Random pairs / triples?

- Random positives
  - Fast
  - Good gradient
- Random negatives
  - Far apart, Small loss, Small gradient
- Pick one negative
  - Closed to each positive



Hard, Semi-hard negatives

- Too noisy
    - No meaningful gradient direction
- Too hard
    - Stronger gradient than positives



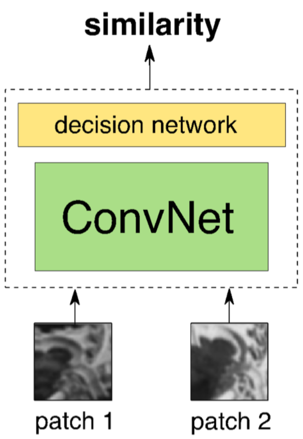
- Semi-hard negatives
- Fine one negatives
    - at same distance as a positive

Applications Matching and Similarity Learning

- Previously features and distance metric have been learned independently
- Use the convolutional descriptors with Nearest neighbor matching with Euclidean distance

  P. Fischer, A. Dosovitskiy, and T. Brox.
  Descriptor matching with convolutional neural networks: a comparison to SIFT.

- Goal: learn descriptors and how to compare patches jointly
- Applications – 2D-3D matching, pose estimation, recognition, retrieval)
- Training - database that contains set of matching patches and nonmatching patches



Zagoruyko, S. and Komodakis, N., 2015. Learning to compare image patches via convolutional neural networks. CVPR 2015

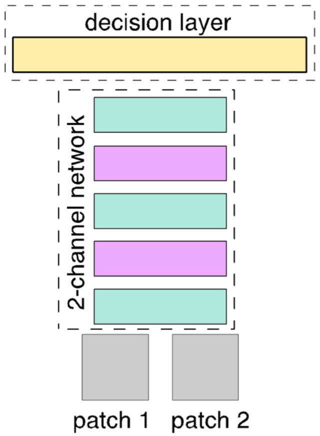
Siamese Networks

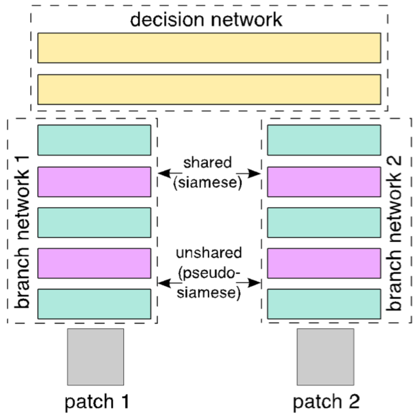
Different way to compare positive and negative examples
Inputs are embedded together – weights are shared
Inputs are embedded independently then merged

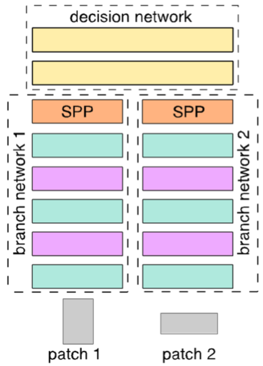
Spatial Pyramid Pool
Layer to be able
To handle patches of
different sizes

Two fully
connected layers

Two channel
image – decision
layer – 1 output
yes or no







Two separate inputs – shared weights
– or pseudo shared weights image –
Outputs are concatenated – passed to
decision layer output yes or no

Zagoruyko, S. and Komodakis, N., Learning to compare image patches via convolutional neural networks, CVPR 2015

Loss Functions

Loss = $\sum$ loss of positive pairs – $\sum$ loss of negative pairs

- Make distance between positive examples small and negative examples large

Patch embedding

- Different Loss for positive pairs $L(x_p, x_q) = \|x_p - x_q\|^2$
- Different Loss for negative pairs (hinge loss)

$$L(x_n, x_q) = \max(0, m^2 - \|x_p - x_q\|^2)$$

Chopra, S., Hadsell, R. and LeCun, Y., 2005, June. Learning a similarity metric discriminatively, with application to face verification. CVPR 2005

- Contrastive Loss

y = {0, 1}

$$L(x_p, x_q, y) = (1 - y).\max(0, m^2 - \|x_p - x_q\|^2) + y\|x_p - x_q\|^2$$

- Combination of positive and negative loss

Network weight regulariztaion

Network output

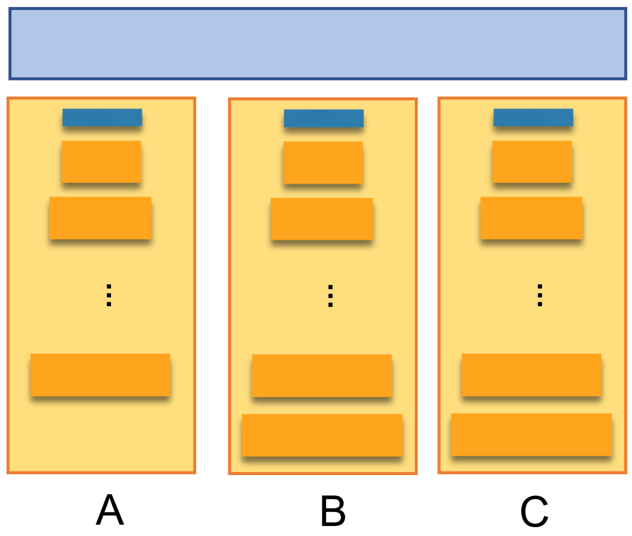
y = {-1, 1}

$$\min_w \frac{\lambda}{2}\|w\|_2 + \sum_{i=1}^{N} \max(0, 1 - y_i o_i^{net})$$

Triplet networks

- Triplet networks – allows ranking of the examples, positive and negative patches in one go

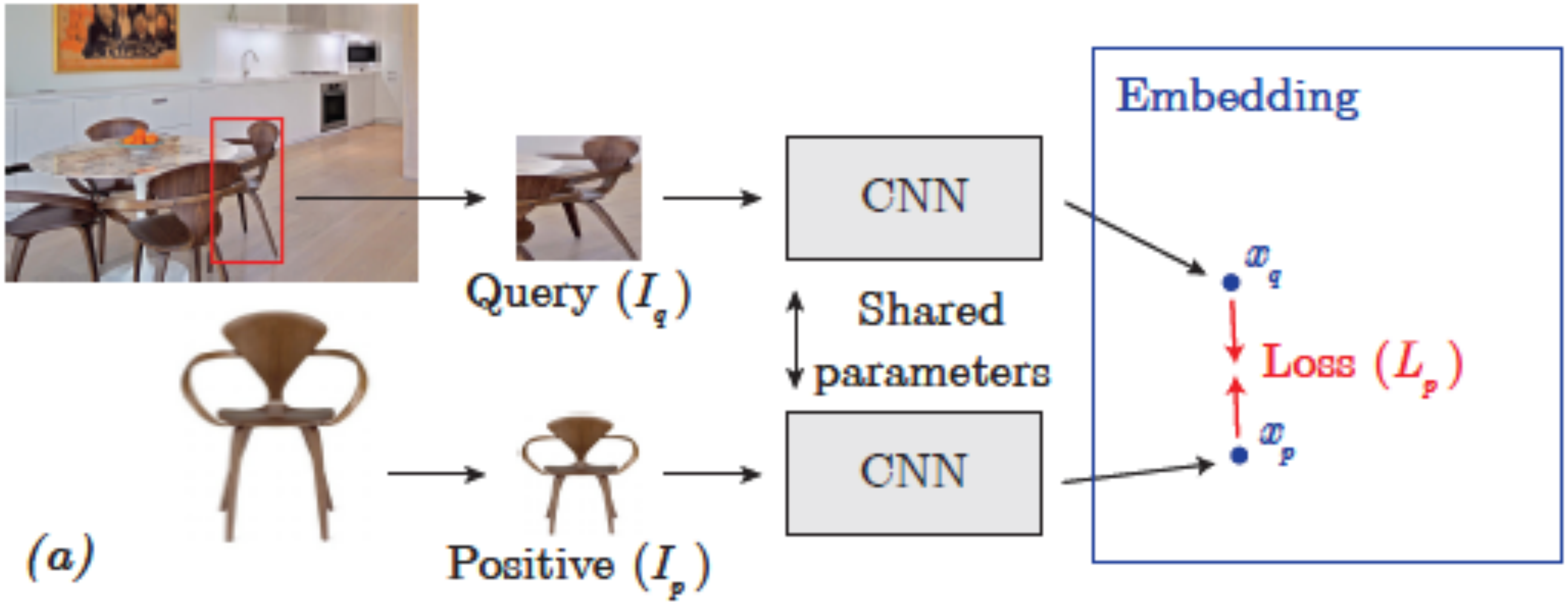
$$L(A, B, C) = max(0, m + D(A, B) – D(A, C))$$



$$\mathbf{D}(f(A), f(B)) < \mathbf{D}(f(A), f(C))$$

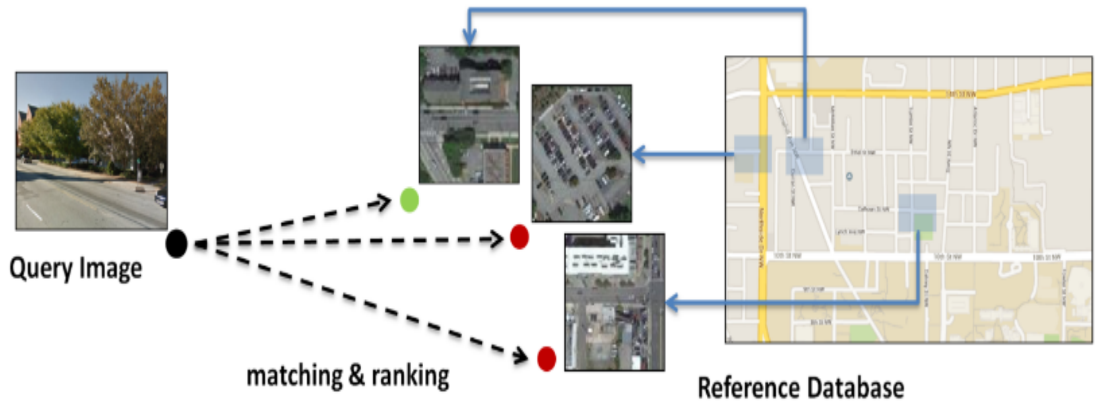
We can use different loss functions for the two types of input pairs.

- Typical positive pair (x_p, x_q) loss: $L(x_p, x_q) = ||x_p - x_q||^2$
(Euclidian Loss)
- Typical negative pair (x_n, x_q) loss :
$L(x_n, x_q) = \max(0, m^2 - ||x_n - x_q||^2)$ (Hinge Loss)



Bell, S. and Bala, K., 2015. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, *34*(4), p.98.

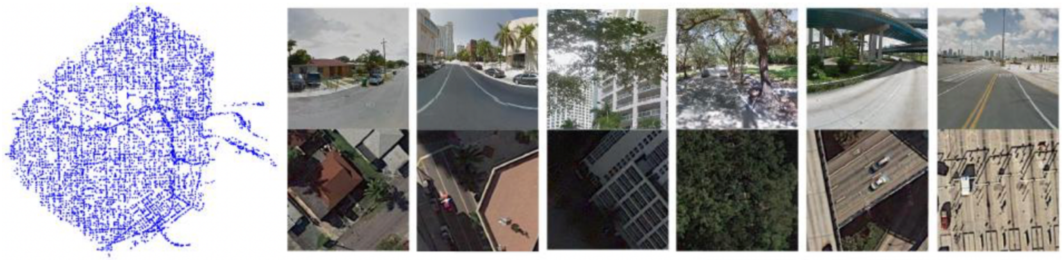
Applications



Vo, N.N. and Hays, J., 2016, October. Localizing and orienting street views using overhead imagery. In European Conference on Computer Vision (pp. 494-509).

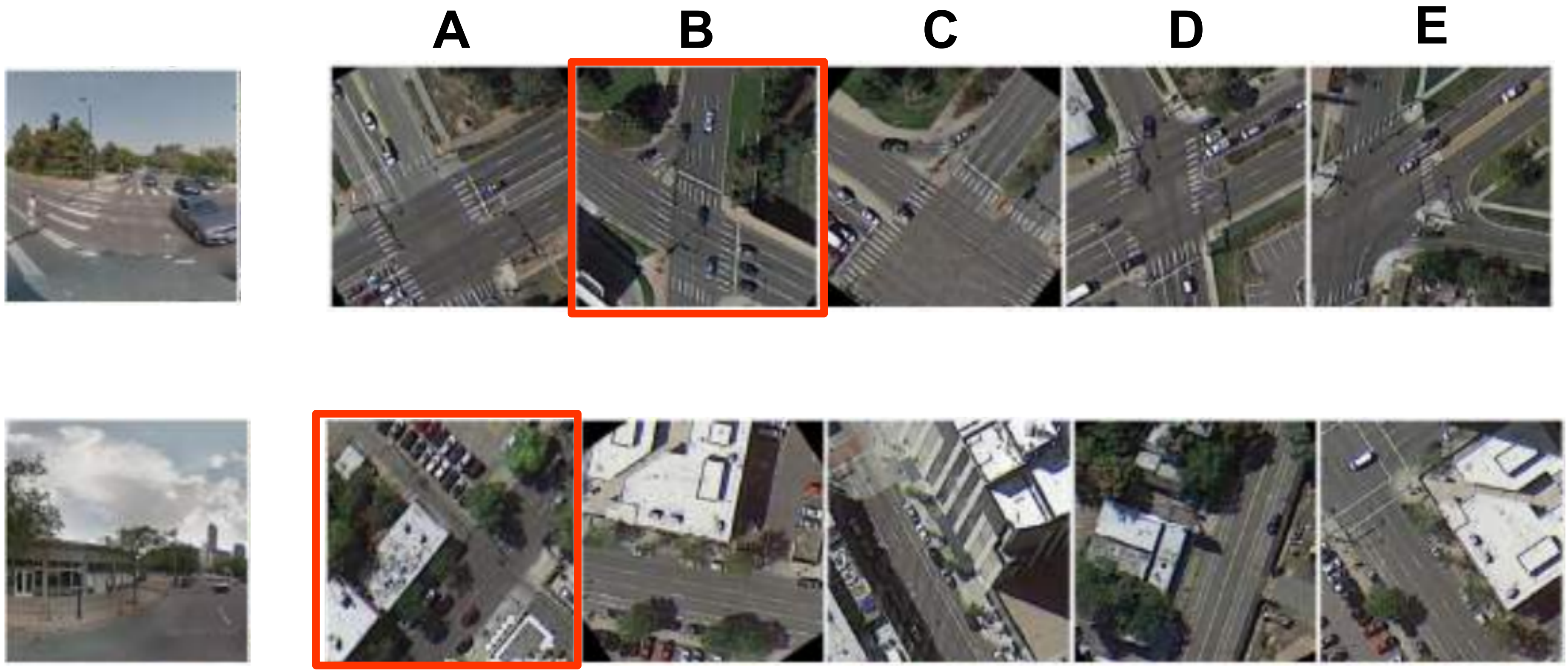
Applications

- Learning discriminative patches from multiple views
- Training data generated from multiple views
- Matching Aerial views to ground level views



Vo, N.N. and Hays, J., 2016, October. Localizing and orienting street views using overhead imagery. In European Conference on Computer Vision (pp. 494-509).

- Query image – determine correct match



Vo, N.N. and Hays, J., 2016, October. Localizing and orienting street views using overhead imagery. In European Conference on Computer Vision (pp. 494-509).

Classification CNN:

$$L(A, B, l) = LogLossSoftMax(f(I), l)$$

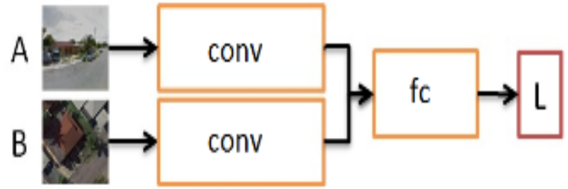
$I = concatenation(A, B)$
$f = AlexNet$
$l = \{0, 1\}$, *label*

Siamese-like CNN:

$$L(A, B, l) = l * D + (1- l) * max(0, m - D)$$

$D = ||f(A) - f(B)||_2$
m = *margin parameter*

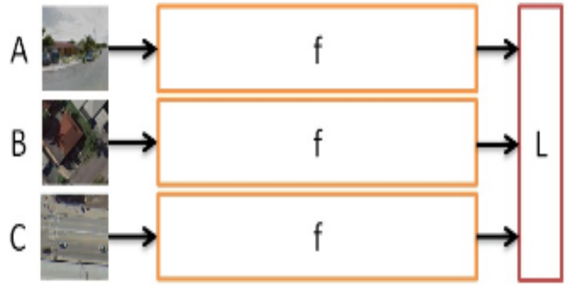
Siamese-classification hybrid network:



$$L(A, B, l) = LogLossSoftMax(f_{fc}(I_{conv}), l)$$

$I_{conv} = concatenation(f_{conv}(A), f_{conv}(B))$

Triplet network CNN:



$$L(A, B, C) = max(0, m + D(A, B) - D(A, C))$$

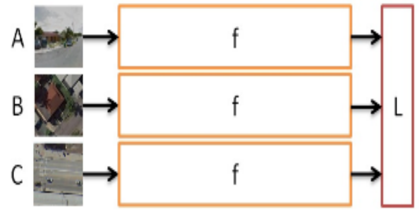
(A, B) is a match pair
(A, C) is a non-match pair

Matching accuracy

Test set	Denver	Detroit	Seattle
Siamese	85.6	83.2	82.9
Triplet	**88.8**	**86.8**	**86.4**

Siamese-like CNN:

Triplet network CNN:



Observation 1:

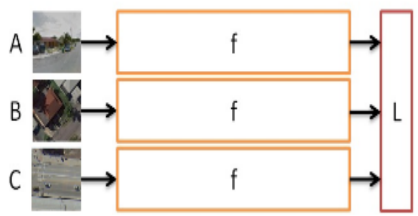
- Triplet network outperforms the Siamese by a large margin

Matching accuracy

Test set	Denver	Detroit	Seattle
Siamese	85.6	83.2	82.9
Siamese-DBL	**90.0**	**88.0**	**88**
Triplet	88.8	86.8	86.4
Triplet-DBL	**90.2**	**88.4**	**87.6**

Siamese-like CNN:

Triplet network CNN:



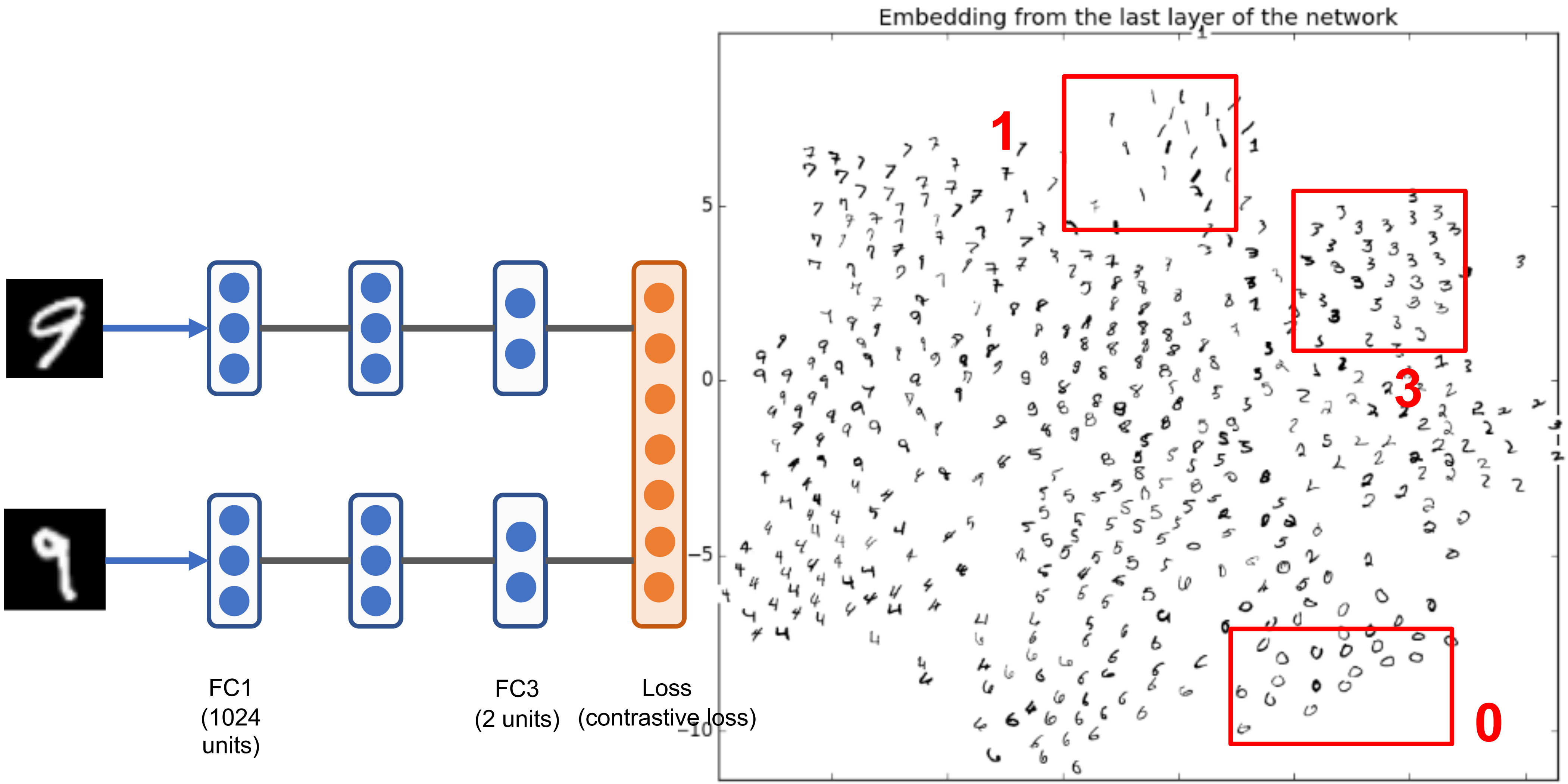
Distance-based logistic (DBL) loss:

$$p(A,B) = \frac{1 + exp(-m)}{1 + exp(D - m)}$$

$$L(A, B, l) = LogLoss\ (p(A, B), l)$$

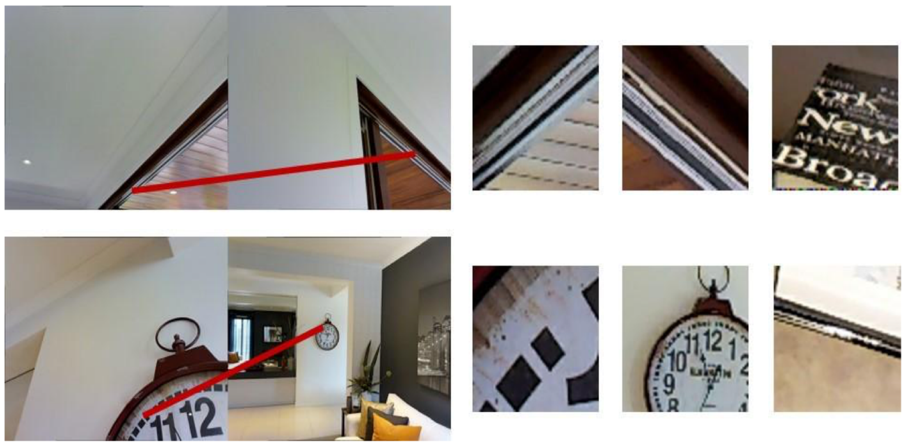
Observation 2:

- Distance-based logistic (DBL) Nets significantly outperform the original network.

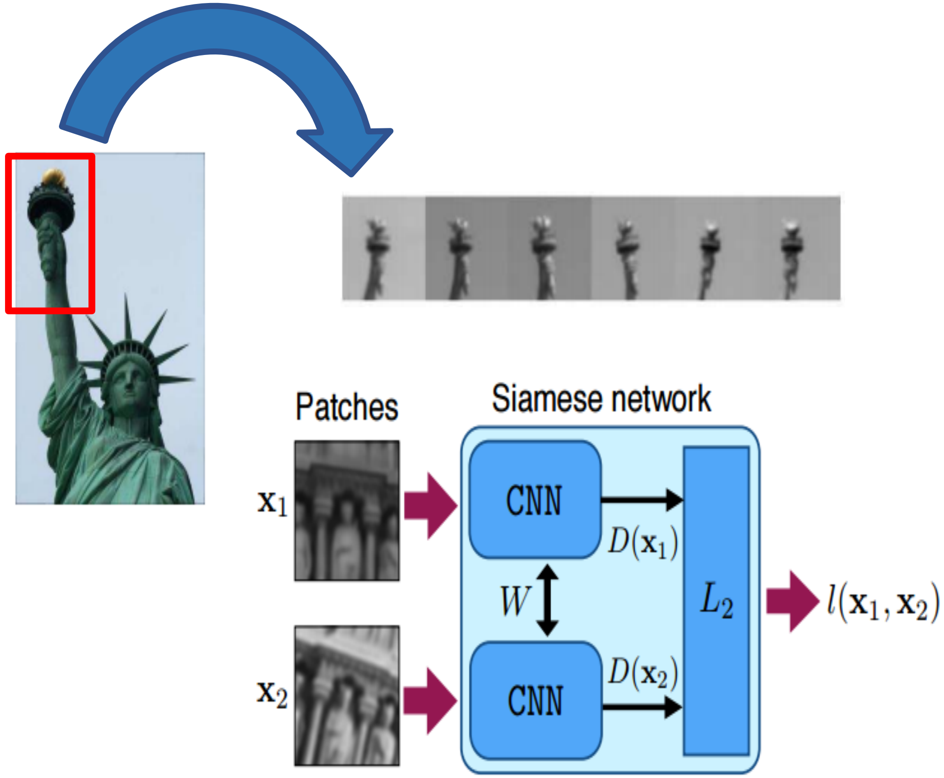

FC1
(1024
units)
FC3
(2 units)
Loss
(contrastive loss)
Embedding from the last layer of the network
1
3
0
5
0
−5
−10
−6
−4
−2
0
2
4
6

Learning Correspondences

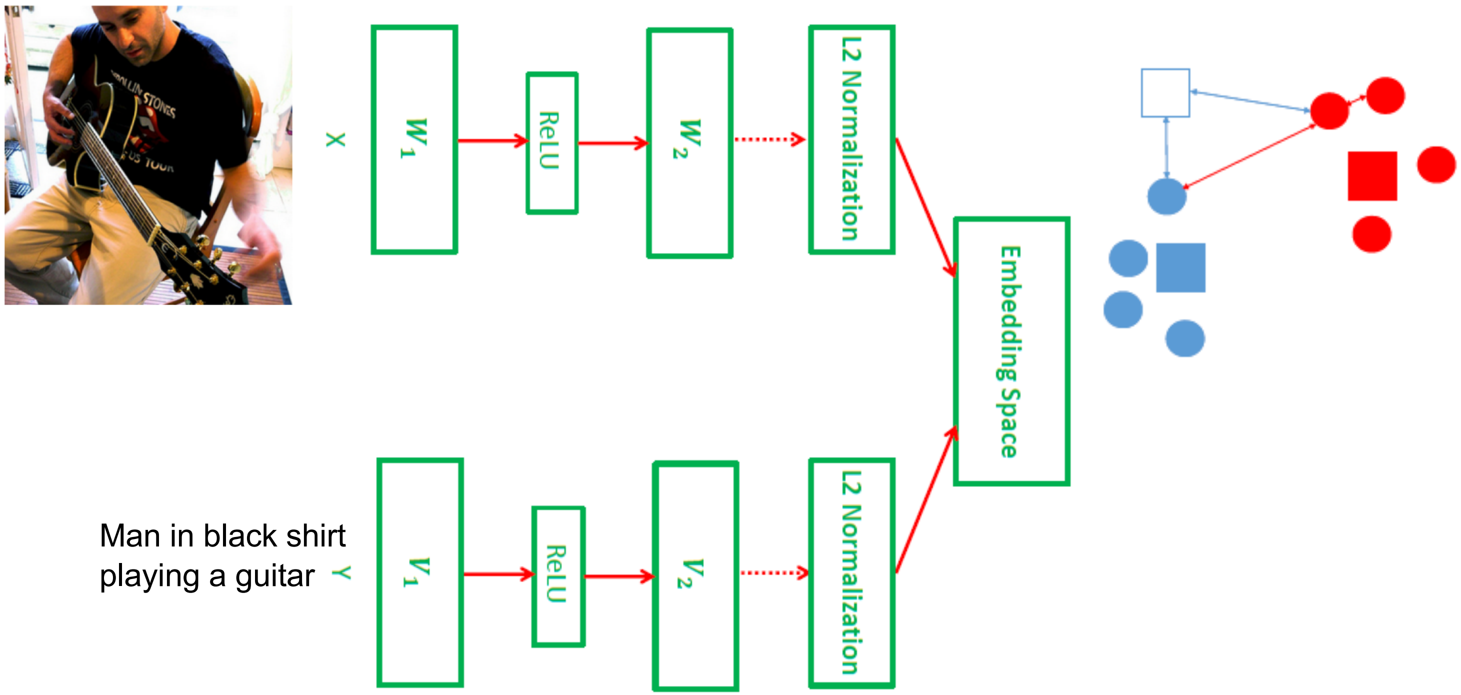
Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P. and Moreno-Noguer, F., 2015. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 118-126).



Matterport3D: Learning from RGB-D Data in Indoor Environments
Angel Chang et. al. Princeton University, Stanford University, Technical University of Munich



Cross modal embeddings



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).

Person Re-indentification problem

Subramaniam, A., Chatterjee, M. and Mittal, A., 2016. Deep Neural Networks with Inexact Matching for Person Re-Identification. In *Advances in Neural Information Processing Systems* (pp. 2667-2675).

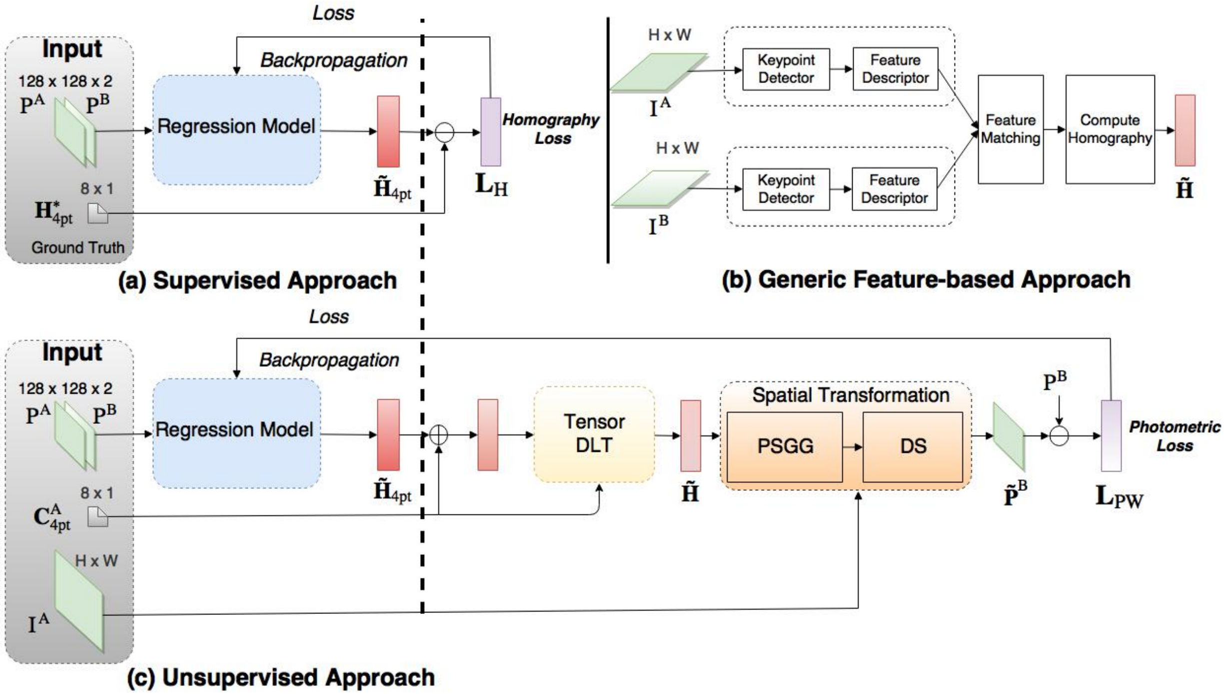
Baseline:

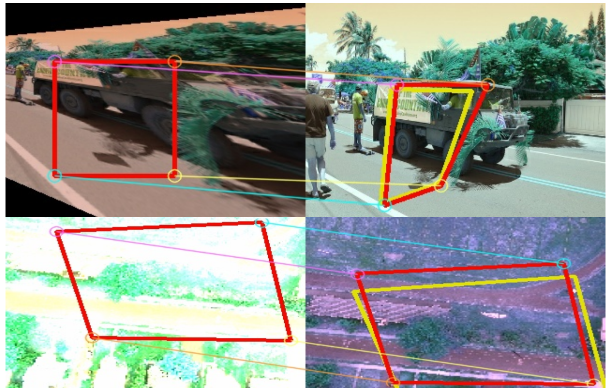
Proposed Method:

Subramaniam, A., Chatterjee, M. and Mittal, A., 2016. Deep Neural Networks with Inexact Matching for Person Re-Identification. In *Advances in Neural Information Processing Systems* (pp. 2667-2675).

Homography Estimation

- Some supervisory signal is easily attainable – matches gathered by SFM and data augmentation techniques





Photometric Loss

$$\mathbf{L}_{PW} = \frac{1}{|\mathbf{x}_i|} \sum_{\mathbf{x}_i} |I^A(\mathscr{H}(\mathbf{x}_i)) - I^B(\mathbf{x_i})|$$

Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model
Ty Nguyen∗, Steven W. Chen∗, Shreyas S. Shivakumar, Camillo J. Taylor, Vijay Kuma