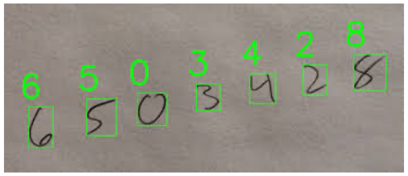
Metric Learning
"How to match things?"

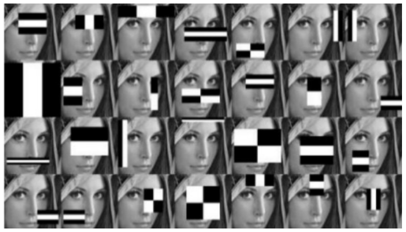
[Slides from Moitreya Chatterjee and Yunan Luo at UIUC]

Applications for Similarity Measures

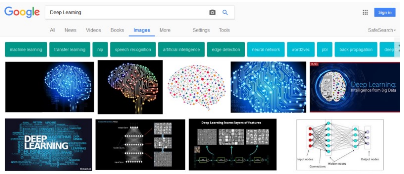
- Recognizing a person's handwriting



- Face identification.



- Search engines: matching a **query** w/ **index**



Questions

Could we solve this as a classification problem?

What happens if a new element is added to index?

What is the network complexity w.rt. |index|?

Outline

- Metric Learning as a measure of Similarity
- Traditional Approaches for Matching
- Challenges with Traditional Matching Techniques
- Deep Learning as a Potential Solution
- Application of Siamese Network for different tasks

Outline

- **Metric Learning as a measure of Similarity**
  - **Notion of a metric**
  - **Unsupervised Metric Learning**
  - **Supervised Metric Learning**
- Traditional Approaches for Matching
- Challenges with Traditional Matching Techniques
- Deep Learning as a Potential Solution
- Application of Siamese Network for different tasks

Notion of a Metric

- A metric is a function that quantifies a distance between every pair of elements in a set, thus inducing a measure of similarity.
- A metric **d(x, y)** must satisfy the following properties ∀ x, y, z:
  - *Non-negativity*: d(x, y) ≥ 0
  - *Identity of discernible*: d(x, y)=0 ⟺ x=y
  - *Symmetry*: d(x, y) = d(y, x)
  - *Triangle Inequality*: d(x, z) ≤ d(x, y) + d(y, z)
- **Hint: recall Euclidean metric** d(x, y) **=** $|\boldsymbol{x} - \boldsymbol{y}|_2$

Types of Metrics

- **Pre-defined Metrics**: Metrics which are fully specified without the knowledge of data.
- e.g., squared Euclidian: $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y})$

- **Learned Metrics**: defined w.r.t. **data**
  - **Unsupervised**: <u>unlabeled</u> data
  - **Supervised**: <u>labeled</u> data

UNSUPERVISED METRIC LEARNING

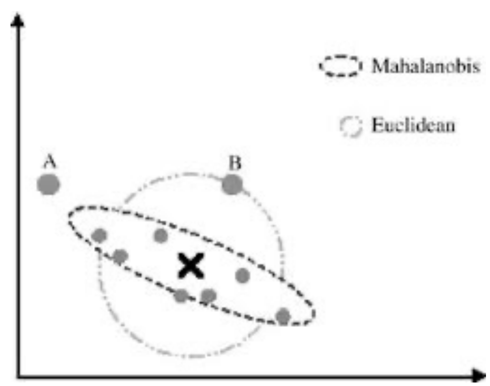
non-neural

Mahalanobis Distance

- Mahalanobis Distance weighs the Euclidian distance between two points, by the standard deviation of the data.
  - $f(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})$; where Σ is the mean-subtracted covariance matrix of all data points.

$$\mathbf{x}^p = \{x_1^p, x_2^p, \cdots x_n^p\}$$

$$\Sigma_{ij} = \sum_p (x_i^p - \overline{x_i})(x_j^p - \overline{x_j})/N$$



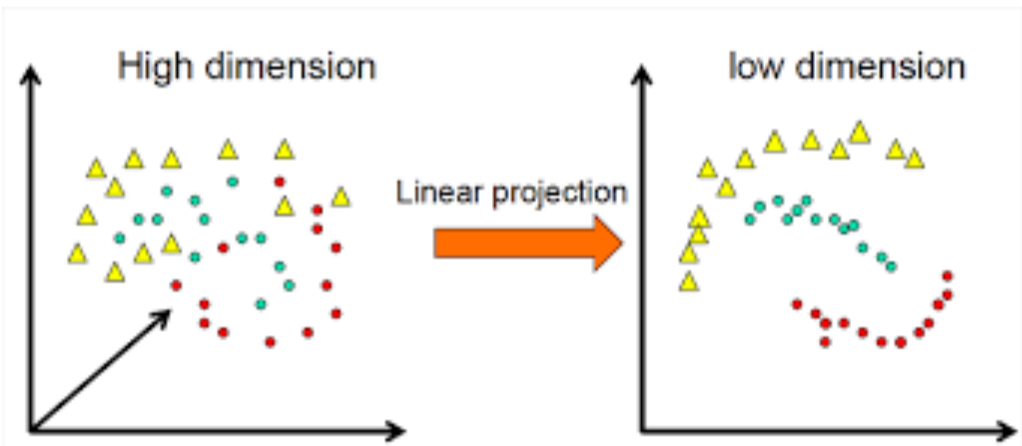
Chandra, M.P., 1936. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India*

SUPERVISED METRIC LEARNING

non-neural

Supervised Metric Learning

- We have access to **labeled** data samples {(x, y)}

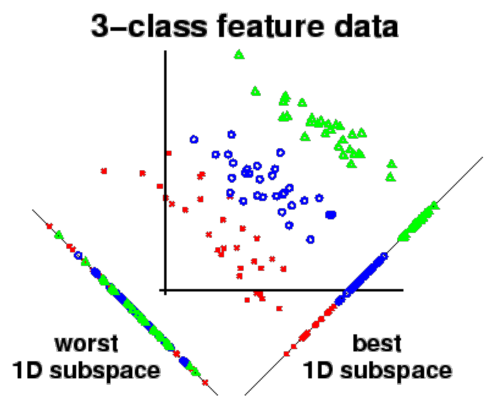


Bellet, A., Habrard, A. and Sebban, M., 2013. A survey on metric learning for feature vectors and structured data. *arXiv*

Linear Discriminant Analysis (Fisher-LDA)

- Project the data to a space to maximize the ratio of "**between class covariance**" /"**within class covariance**"

- This is given by: $E(w) = \max_w (w^T S_B w)/(w^T S_W w)$



Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics*

Linear Discriminant Analysis (Fisher-LDA)

- Compute the Mean for Each Class
- Compute the **Between**-Class **Scatter** Matrix S_b
- Compute the **Within**-Class **Scatter** Matrix S_w
- Solve the (generalized) Eigenvalue problem $S_w^{-1} S_b$
- $\boldsymbol{w}$ is eigenvector with largest eigenvalue
- Project each data point to $\tilde{x}_i = \boldsymbol{w} \cdot x_i$
- Use any technique for classification of linearly separable data

$$S_B = \sum_c (\boldsymbol{\mu}_c - \bar{\mathbf{x}})(\boldsymbol{\mu}_c - \bar{\mathbf{x}})^T$$

$$S_W = \sum_c \sum_{i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T$$

Linear Discriminant Analysis (Fisher-LDA)

- Assumptions
  - The data for each class follows a Gaussian distribution
  - All classes have the same covariance matrix
  - The classes are (to some extent) linearly separable

- Applications
  - Face Recognition (e.g., “FisherFaces“)
  - Medical Diagnosis (classifying diseases)
  - Financial Analysis (predicting market trends)
  - Text Classification (spam detection, sentiment analysis)

Outline

- Metric Learning as a measure of Similarity
- **Traditional Approaches for Matching**
- **Challenges with Traditional Matching Techniques**
- **Deep Learning as a Potential Solution**
- Application of Siamese Network for different tasks

Traditional Approaches for Matching

The traditional approach for matching images, relies on the following pipeline:

1. **Extract Features**: e.g. color histograms of the images
2. **Learn Similarity**: e.g. L_1-norm on features, or SVM

Stricker, M.A. and Orengo, M., 1995, March. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology* (pp. 381-392). International Society for Optics and Photonics.

Challenges with Traditional Matching Techniques

The traditional approach for matching images, relies on the following pipeline:

1. **Extract Features**: e.g. color histograms of the images
2. **Learn Similarity**: e.g. L_1-norm on features, or SVM

Problems

1 is a hand-crafted pipeline

1 and 2 are separate

Stricker, M.A. and Orengo, M., 1995, March. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology* (pp. 381-392). International Society for Optics and Photonics.

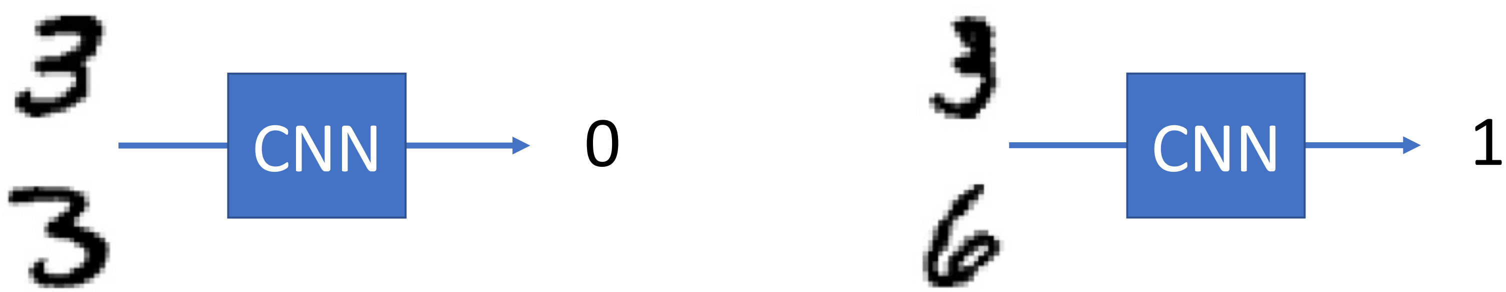
Deep Learning to the Rescue!

- CNNs can <u>**jointly**</u> **optimize** (i.e., end-to-end learning)

1. "Extract Features" (via CNN)
2. "Learn Similarity" (via CNN)

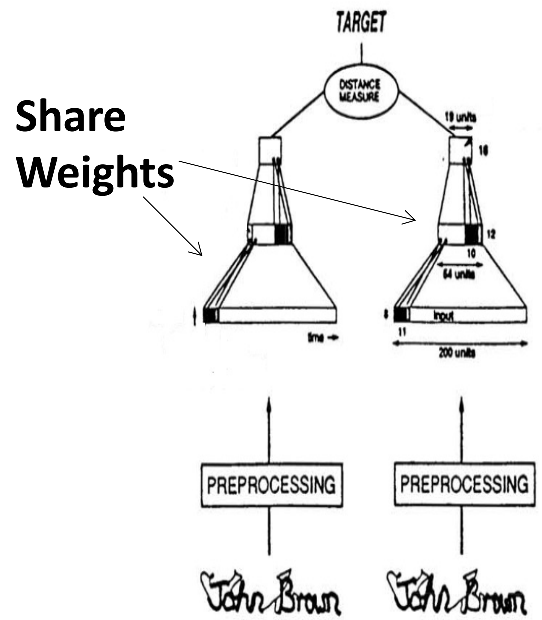
Revisit the Problem

- **Input**: pair of input images
- **Output**: can take a variety of forms…
  - A binary label: 0 (same) or 1 (different)
  - A real number: how similar a pair is



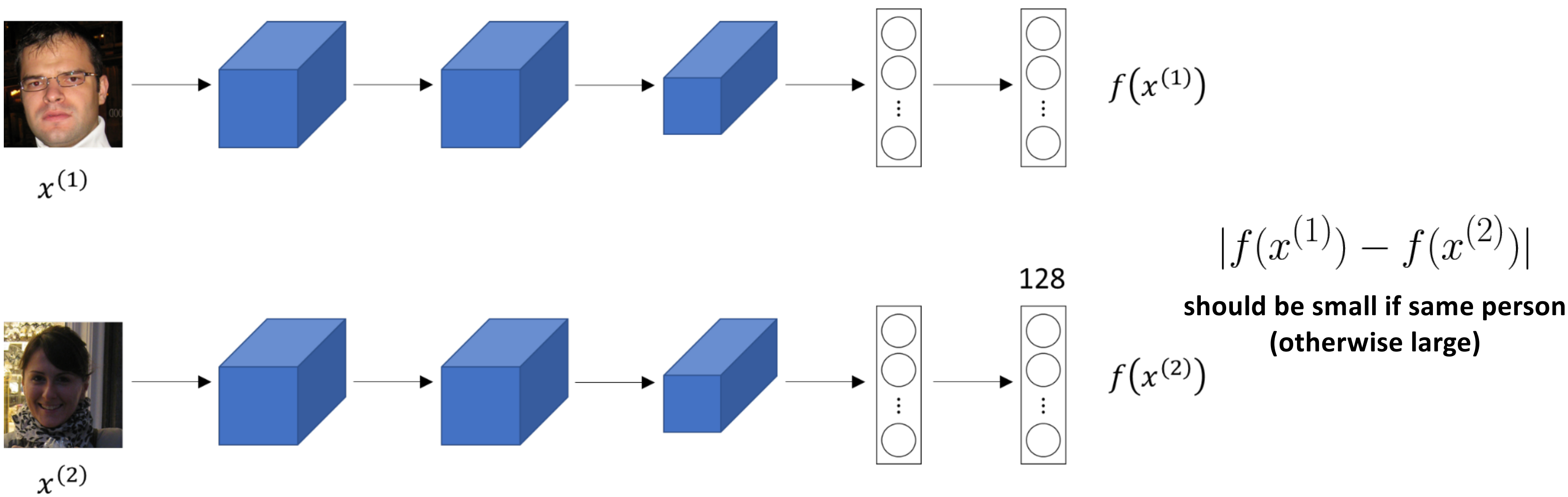
Typical Siamese CNN

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



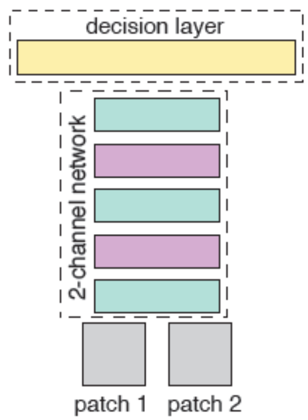
Bromley et al., 1993. Signature Verification Using A "Siamese" Time Delay Neural Network. *IJPRAI*, *7*(4), pp.669-688.

Standard architecture of Siamese CNN

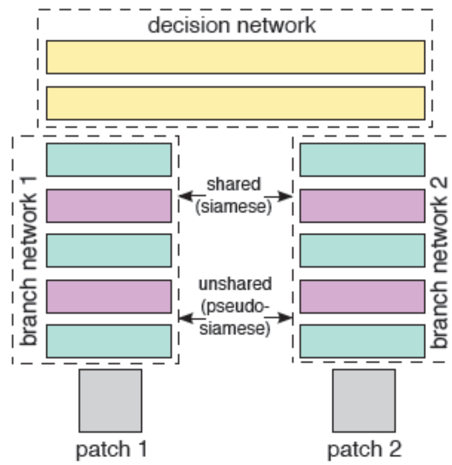


Siamese CNN – Variants

No one "architecture" fits all! Design largely governed by what performs well empirically on the task at hand.



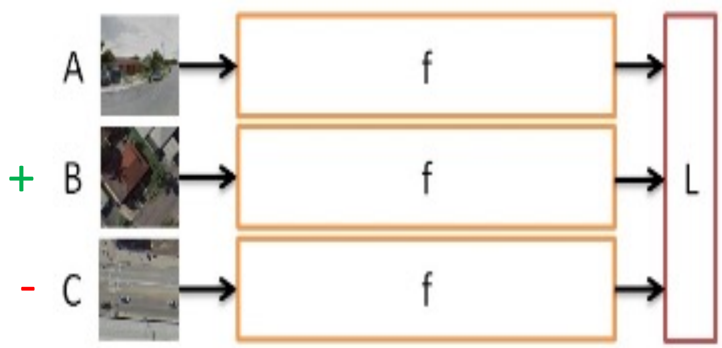
Inputs are merged right at the onset



Inputs are first embedded independently, then merged.

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

Siamese CNN – Triplet Network

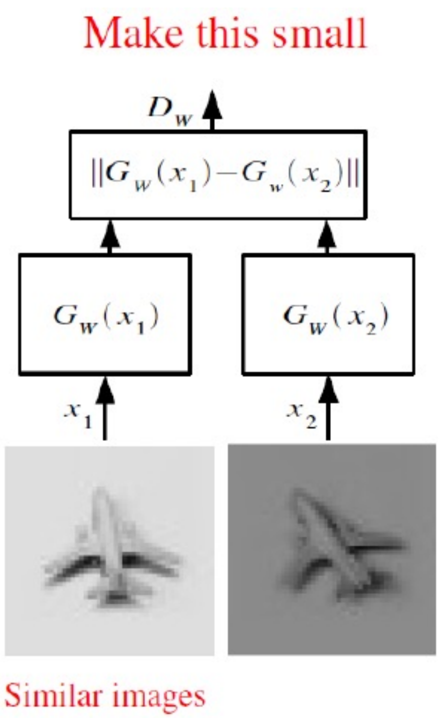


$d(f(A), f(B)) < d(f(A), f(C))$

- Compare triplets in one go: check if the sample in the **topmost** channel, is more like the one in the middle or the one in the bottom.
- Allows us to learn ranking between samples.

Vo, N.N. and Hays, J., 2016, October. Localizing and orienting street views using overhead imagery. ECCV

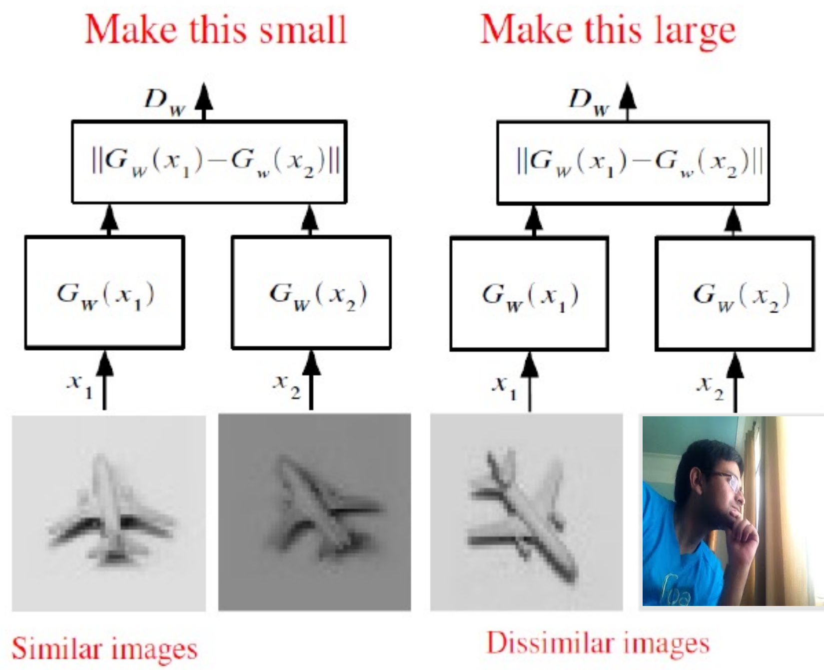
Siamese CNN – Loss Function



- **Is there a problem with this formulation?**
  - Yes: **trivial solution** is to embed every input to the same point
  - Every pair becomes a positive pair

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*

Siamese CNN – Loss Function



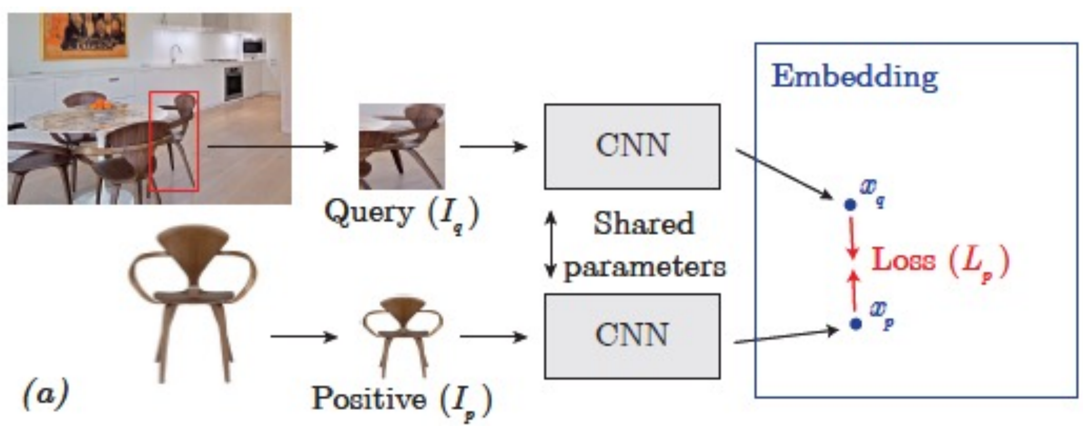
The final loss is defined as:

L = ∑loss of positive pairs + ∑ loss of negative pairs

Siamese CNN – Loss Function

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)

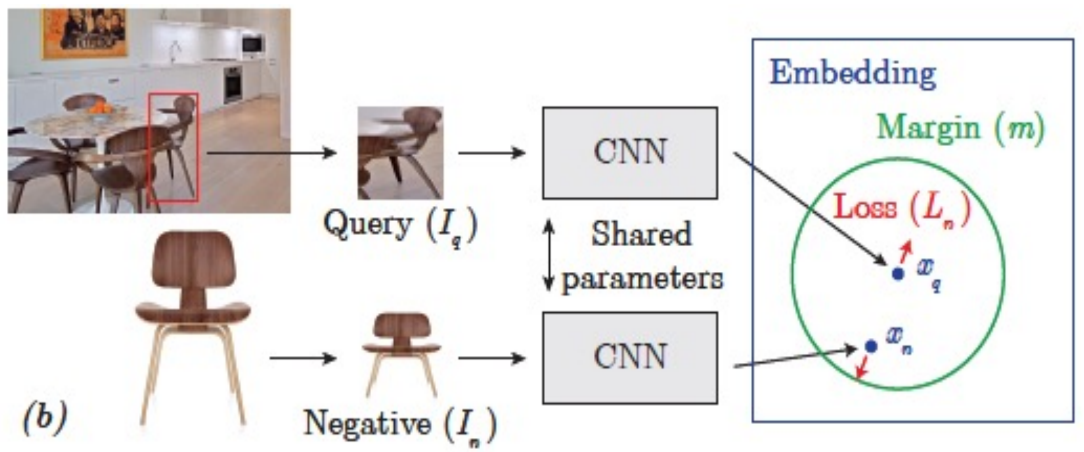


Bell, S. and Bala, K., 2015. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, *34*(4), p.98.

Siamese CNN – Loss Function

- Typical negative pair (x_n, x_q) loss :

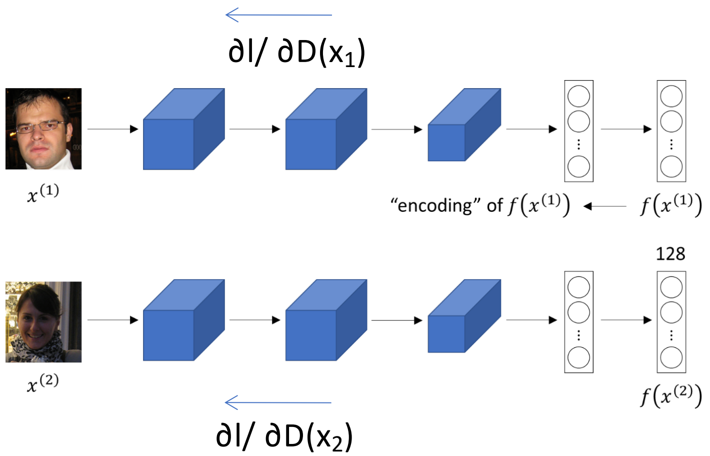
  $L(x_n, x_q) = \max(0, m^2 - ||x_n - x_q||^2)$ (Hinge Loss)



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.

Siamese CNN – Training

- Update each of the two streams independently and then average the weights.

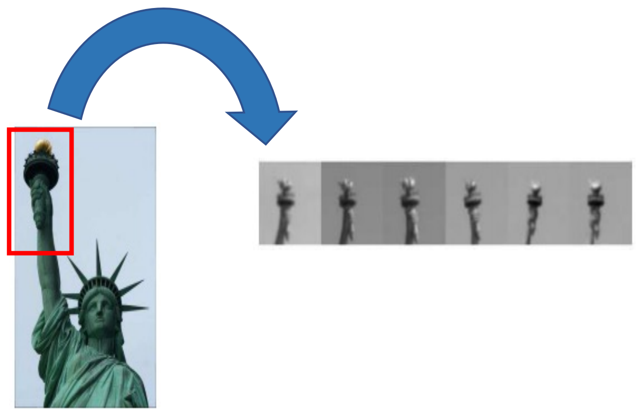


- Data augmentation may be used for more effective training
  - Hallucinate more examples via random crops, flips, etc.

Outline

- Metric Learning as a measure of Similarity
- Traditional Approaches for Matching
- Challenges with Traditional Matching Techniques
- Deep Learning as a Potential Solution
- **Application of Siamese Network for different tasks**
  - Generating invariant and robust descriptors
  - Person re-Identification
  - Rendering a street from different viewpoints
  - Person re-id, viewpoint invariance and multi-modal data
  - Sentence Matching

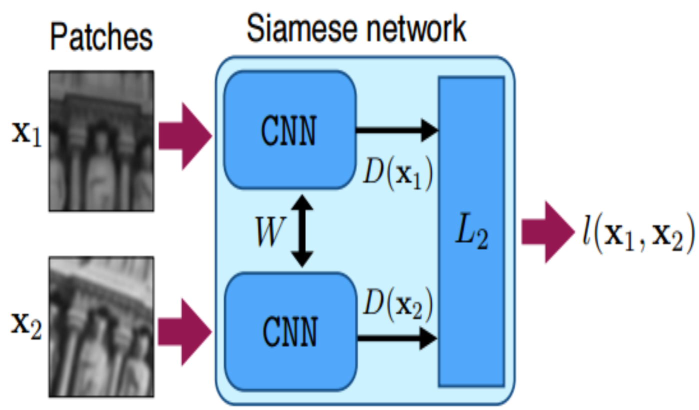
Discriminative Descriptors for Local Patches



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., Discriminative learning of deep convolutional feature point descriptors. ICCV 2015

Deep Descriptor



$$l(x_1, \mathrm{x}_2) = \begin{cases} \left|\left|D(x_1) - D(x_2)\right|\right|_2, & p_1 = p_2 \\ \max\left(0, C - \left|\left|D(x_1) - D(x_2)\right|\right|_2\right), & p_1 \neq p_2 \end{cases}$$

Use the CNN outputs of our Siamese networks as descriptor

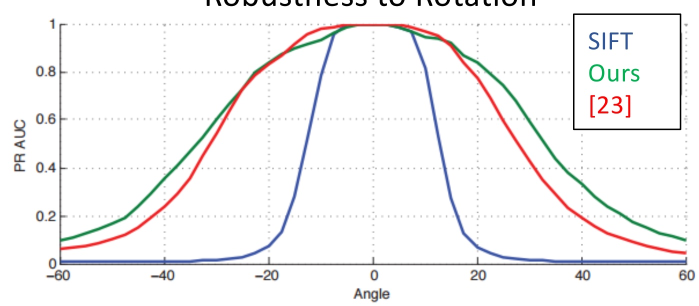
Evaluation

Comparison of area under precision-recall curve

Dataset	SIFT (Non-deep)	[23](Non-deep)	Ours
ND	0.346	0.663	**0.667**
TO	0.425	**0.709**	0.545
LY	0.226	0.558	**0.608**
All	0.370	0.693	**0.756**

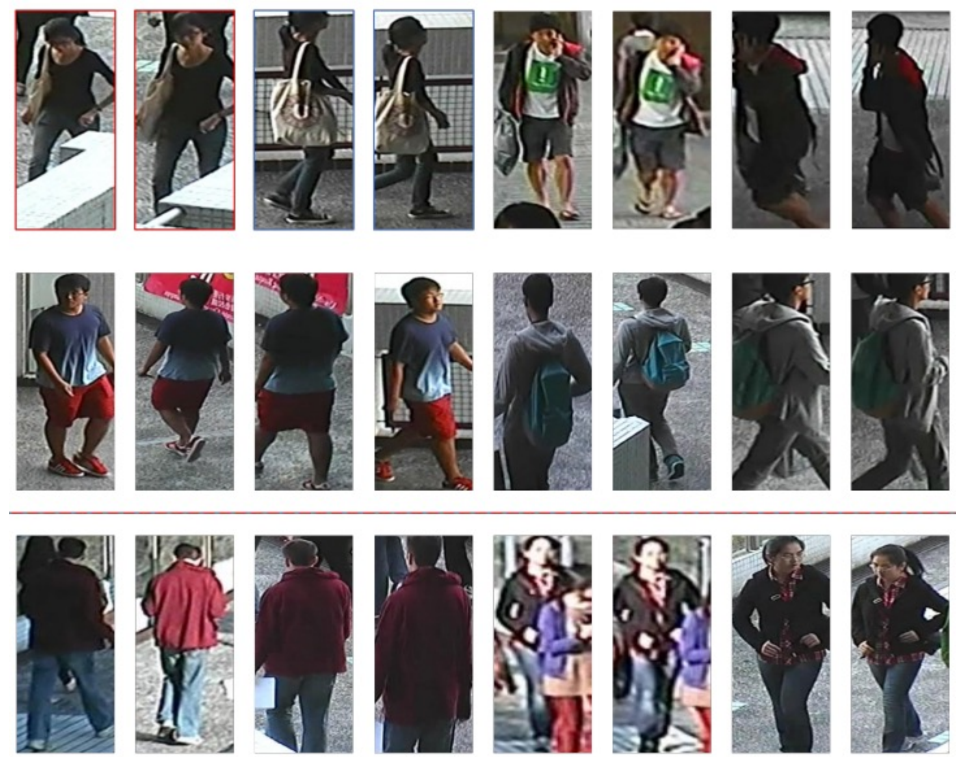
SIFT: hand-crafted features
[23]: descriptor via convex optimization

Robustness to Rotation



Person Re-Identification

The **CUHK03** consists of 14,097 images of 1,467 different identities, where 6 campus cameras were deployed for image collection and each identity is captured by 2 campus cameras. This dataset provides two types of annotations, one by manually labelled bounding boxes and the other by bounding boxes produced by an automatic detector. The dataset also provides 20 random train/test splits in which 100 identities are selected for testing and the rest for training



True
positive

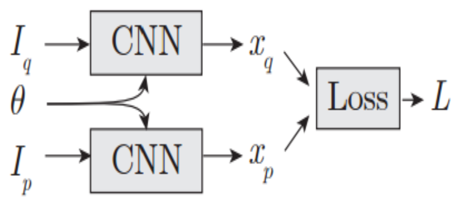
True
negative

Quick Test

Are they the same person?

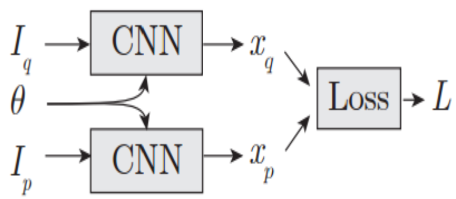
Ahmed, E., Jones, M. and Marks, T.K., 2015. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3908-3916).

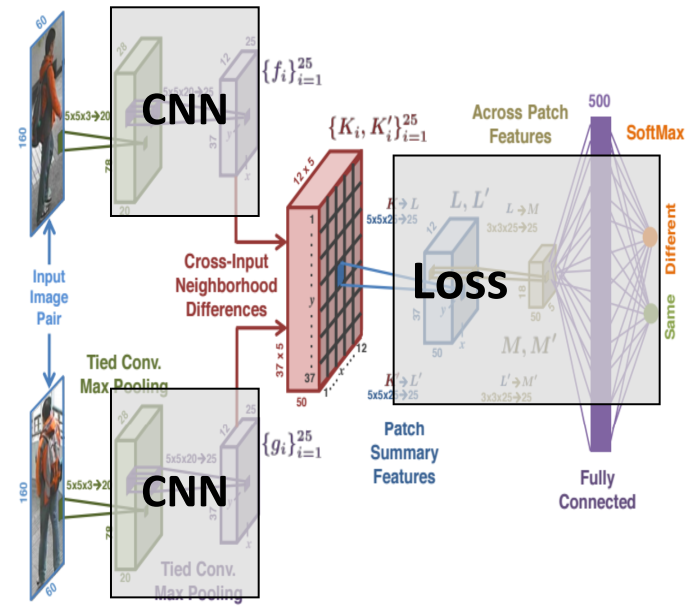
Proposed Architecture



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).

Proposed Architecture

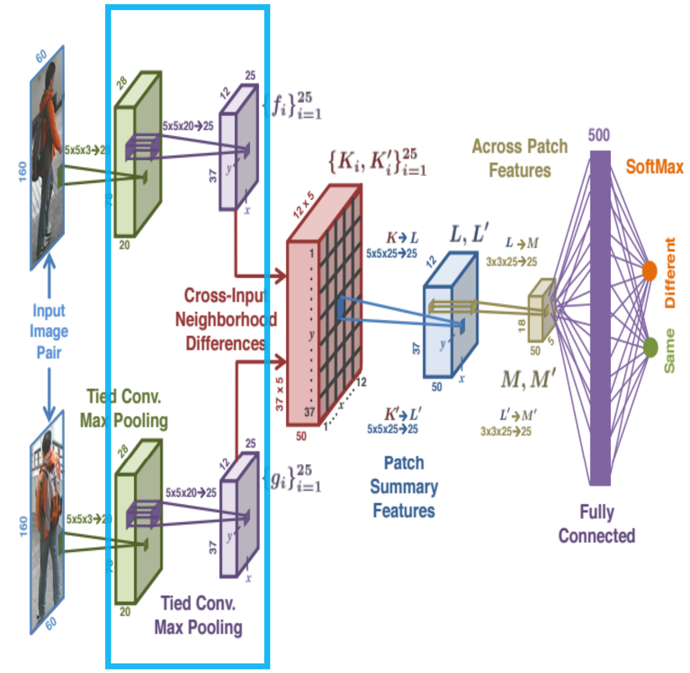




Ahmed, E., Jones, M. and Marks, T.K., 2015. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3908-3916).

Tied Convolution

- Use convolutional layers to compute higher-order features
- Shared weights



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).

Cross-Input Neighborhood Differences

- Compute *neighborhood difference* of two feature maps, instead of elementwise difference.

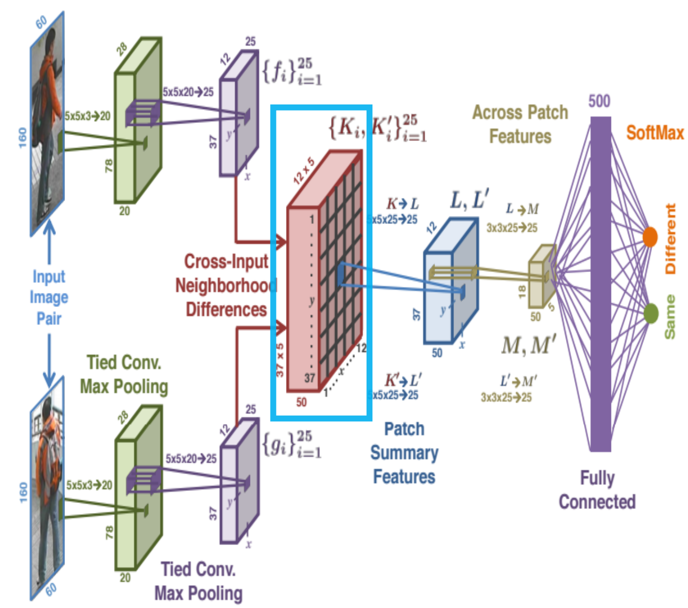
Example: f, g are feature maps of two input images

f

5	7	2
1	4	2
3	4	4

g

1	4	1
2	3	5
1	2	3



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).

Cross-Input Neighborhood Differences

- Compute *neighborhood difference* of two feature maps, instead of elementwise difference.

Example: f, g are feature maps of two input images

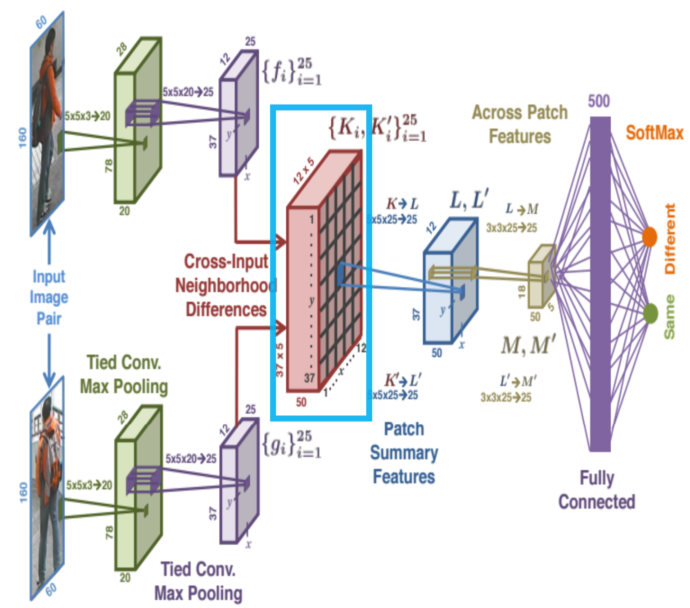
f

5	7	2
1	4	2
3	4	4

g

1	4	1
2	3	5
1	2	3

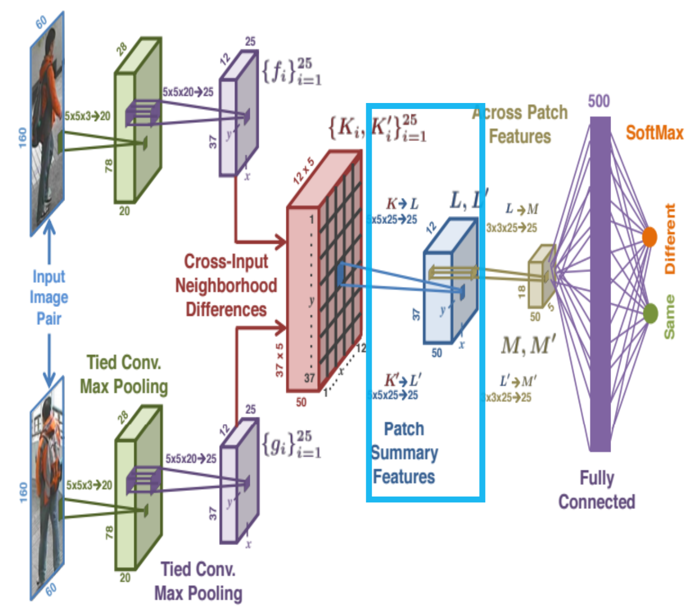
$$K(1,1) = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} - \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 3 & 2 \end{bmatrix}$$



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).

Patch Summary Features

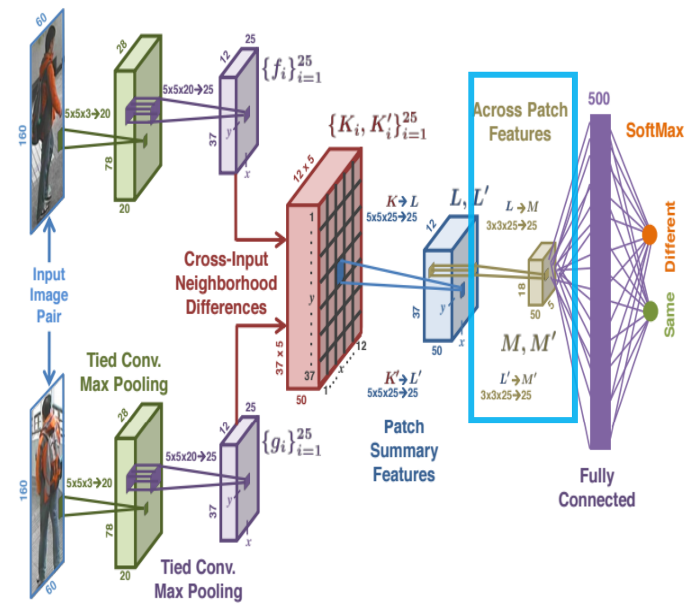
- Convolutional layers with 5x5 filters and stride 5 (the size of neighborhood patch).
- Provides a high-level summary of the cross-input differences in a neighborhood patch.



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).

Across-Patch Features

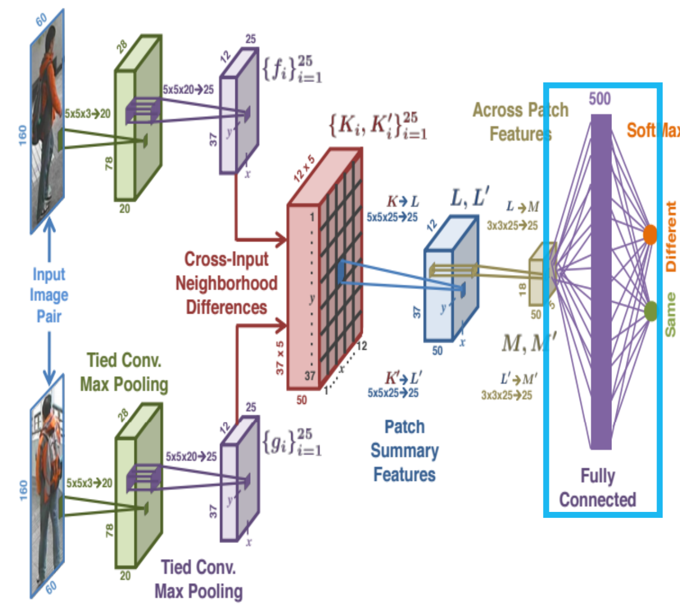
- Convolutional layers with 3x3 filters and stride 1.
- Learn spatial relationships across neighborhood differences



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).

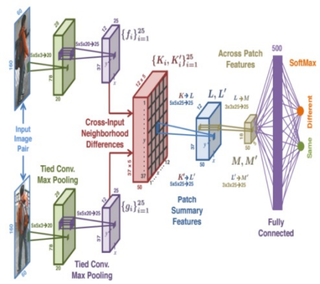
Across-Patch Features

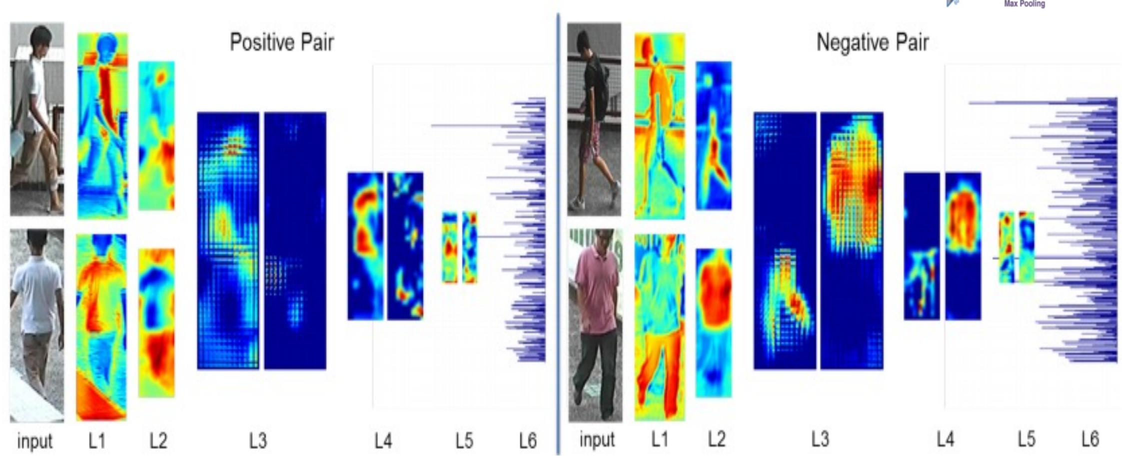
- Fully connected layer.
- Combine information from patches that are far from each other.
- Output: 2 softmax units



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).

Visualization of Learned Features

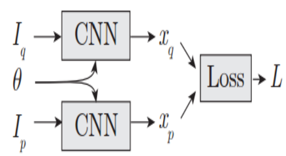


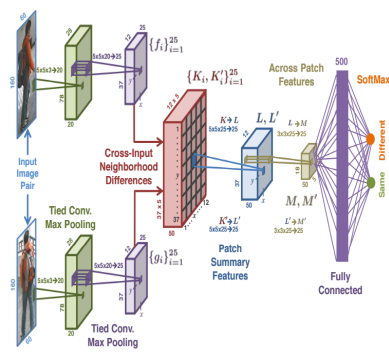


Ahmed, E., Jones, M. and Marks, T.K., 2015. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3908-3916).

Evaluation

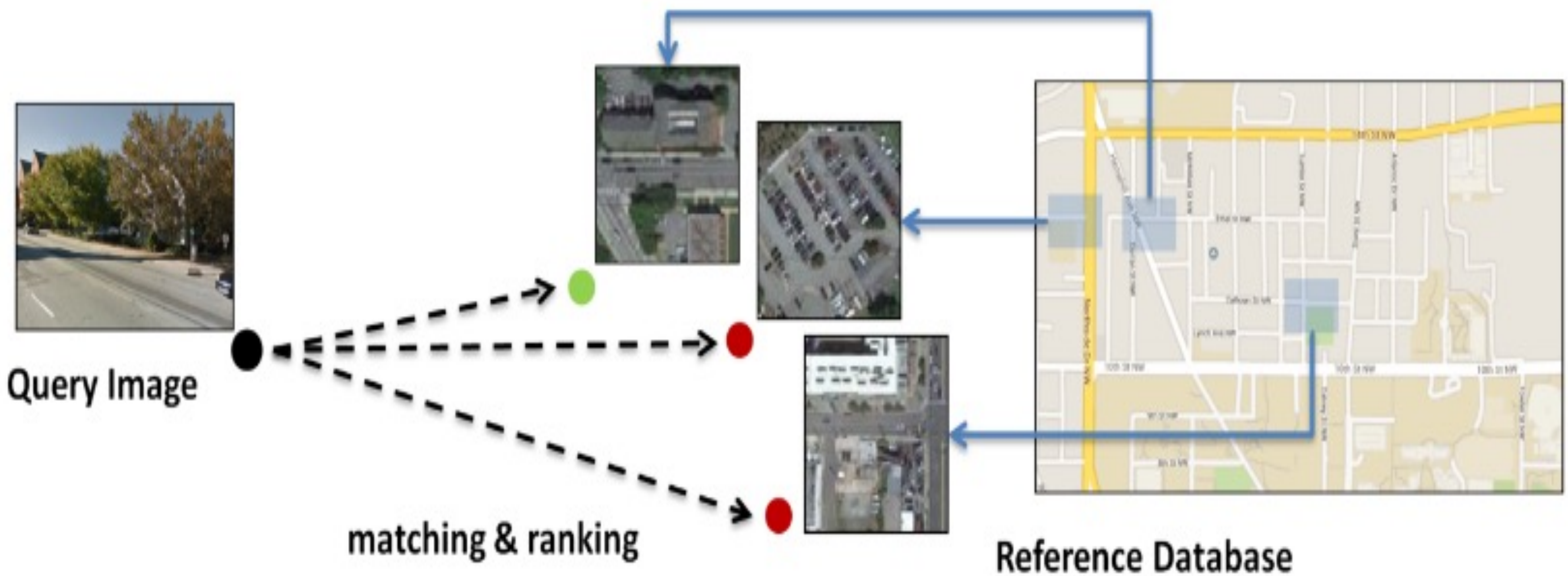
Method	Regular Siamese Network	This work
Identification rate	42.19%	**54.74%**





Ahmed, E., Jones, M. and Marks, T.K.. An improved deep learning architecture for person re-identification. CVPR 2015

Street-View to Overhead-View Image Matching



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).

Street-View to Overhead-View Image Matching

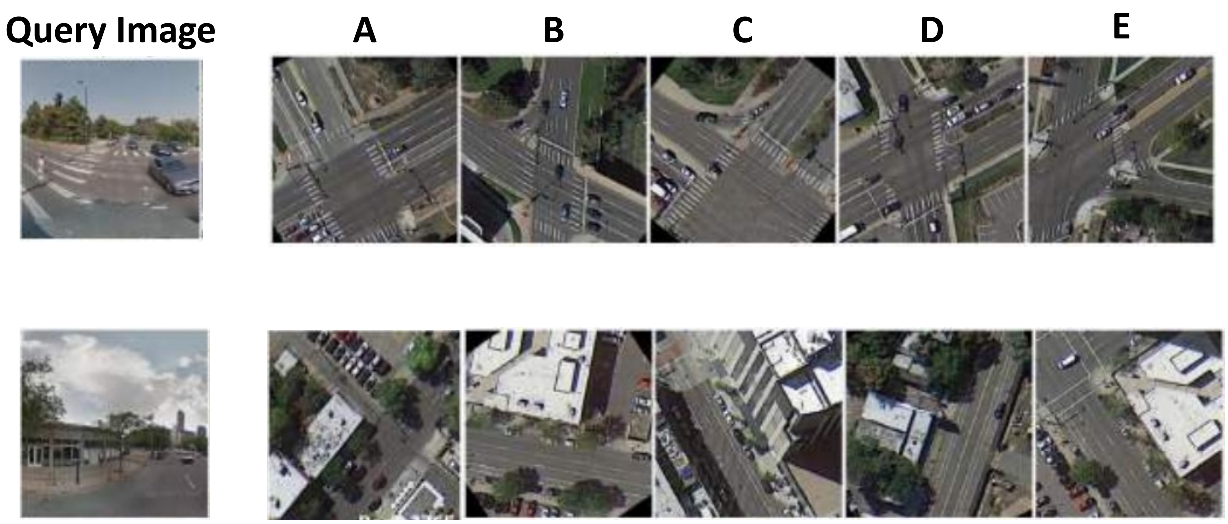
Query:

Matching Image:

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).

Quick Test

Which one is the 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).

Quick Test

Which one is the 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).

CNN Architectures

Classification CNN:

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

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

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).

CNN Architectures

Classification CNN:

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

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

Siamese-like CNN:

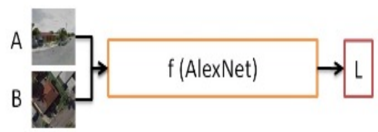
$$L(A, B, l) = l * D + (1- l) * max(0, m - D)$$

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

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).

CNN Architectures

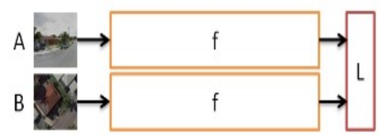
Classification CNN:



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

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

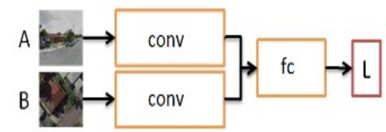
Siamese-like CNN:



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

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

Siamese-classification hybrid network:



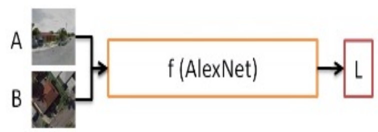
$$L(A, B, l) = LogLossSoftMax(f_{fc}(I_{conv}), l)$$

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

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).

CNN Architectures

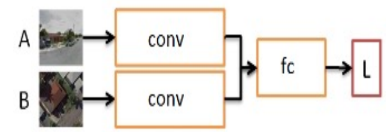
Classification CNN:



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

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

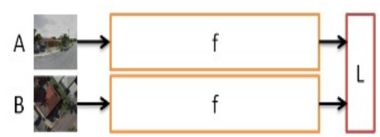
Siamese-classification hybrid network:



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

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

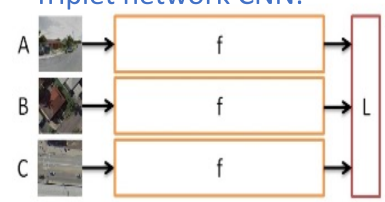
Siamese-like CNN:



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

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

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

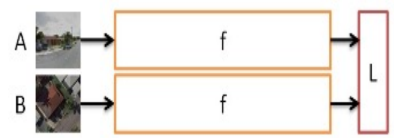
Vo, N.N. and Hays, J., 2016, October. Localizing and orienting street views using overhead imagery. In European Conference on Computer Vision (pp. 494-509).

Performance of Different Networks

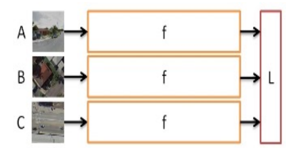
Matching accuracy

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:

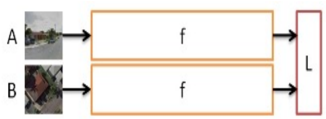
- Triplet network outperforms the Siamese by a large margin

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).

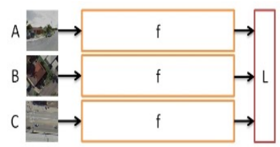
Performance of Different Networks

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.

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).

Performance of Different Networks

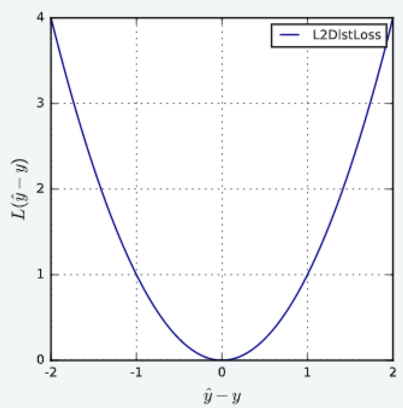
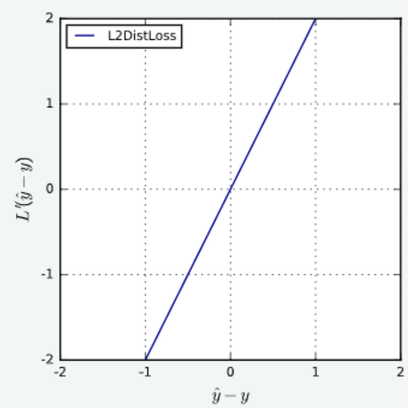
L2DistLoss

`LossFunctions.L2DistLoss` — *Type*.

```
L2DistLoss <: DistanceLoss
```

The least squares loss. Special case of the `LPDistLoss` with `P=2`. It is strictly convex.

source

Lossfunction	Derivative
	
$L(r) = \mid r \mid^2$	$L'(r) = 2r$

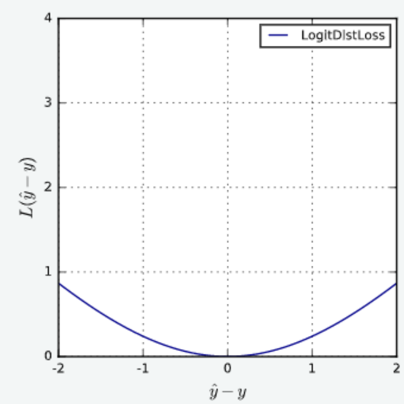
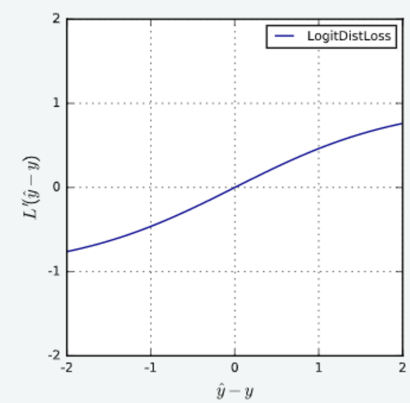
LogitDistLoss

`LossFunctions.LogitDistLoss` — *Type*.

```
LogitDistLoss <: DistanceLoss
```

The distance-based logistic loss for regression. It is strictly convex and Lipschitz continuous.

source

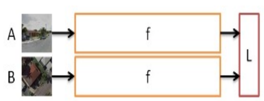
Lossfunction	Derivative
	
$L(r) = -\ln \frac{4e^r}{(1+e^r)^2}$	$L'(r) = \tanh(\frac{r}{2})$

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).

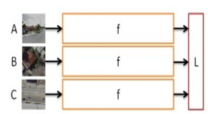
Performance of Different Networks

Test set	Denver	Detroit	Seattle
Siamese Net	85.6	83.2	82.9
Triplet Net	88.8	86.8	86.4
Classification Net	**90.0**	**87.8**	**87.7**
Hybrid Net	**91.5**	**88.7**	**89.4**

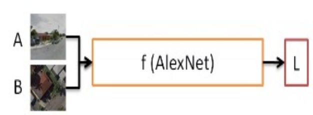
Siamese-like CNN:



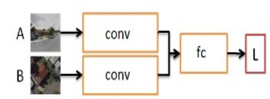
Triplet network CNN:



Classification CNN:



Classification-siamese hybrid:



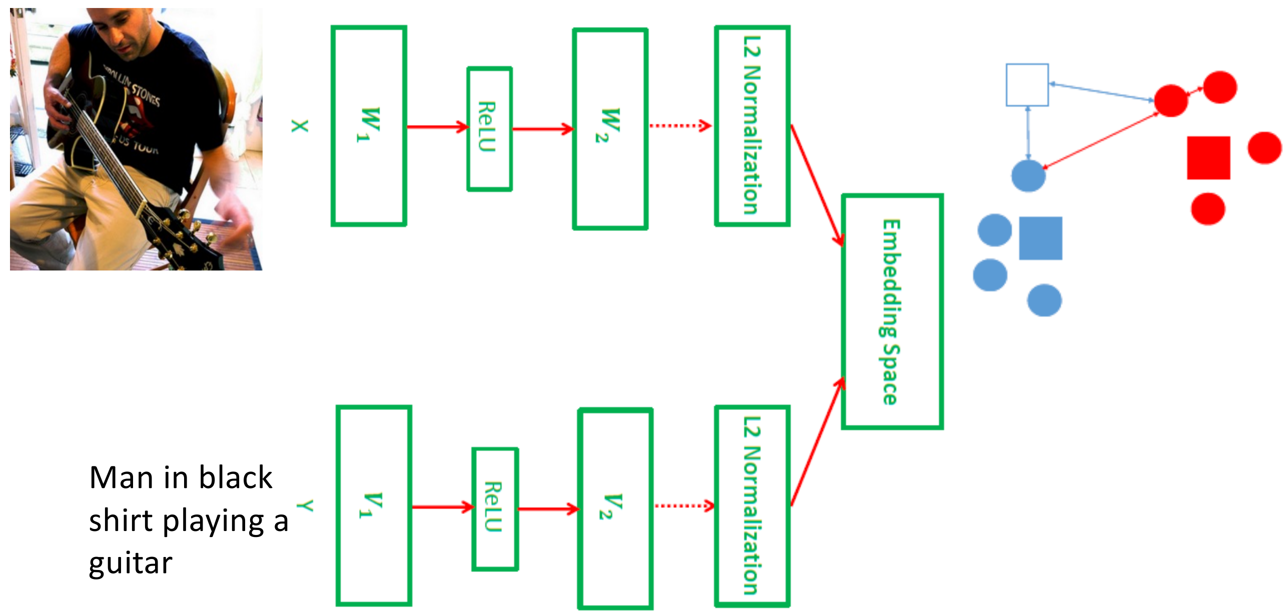
Observation 3:

- Classification networks achieved better accuracy than Siamese and triplet networks.
- Jointly extract and exchange information from both input images.

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).

More applications

CROSS-MODAL EMBEDDING

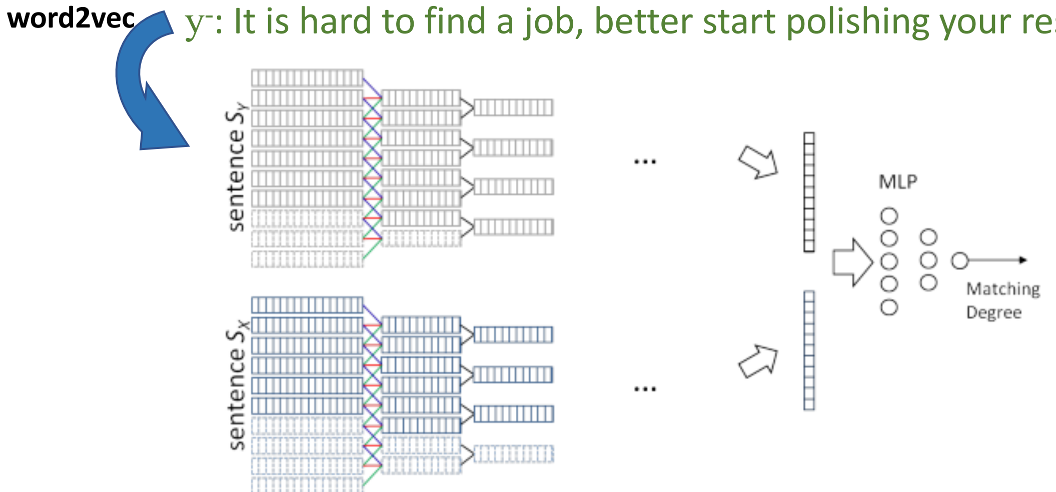


Two stream networks have also been used for cross-modal embedding tasks. Here inputs from different modalities are mapped to a common space.

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).

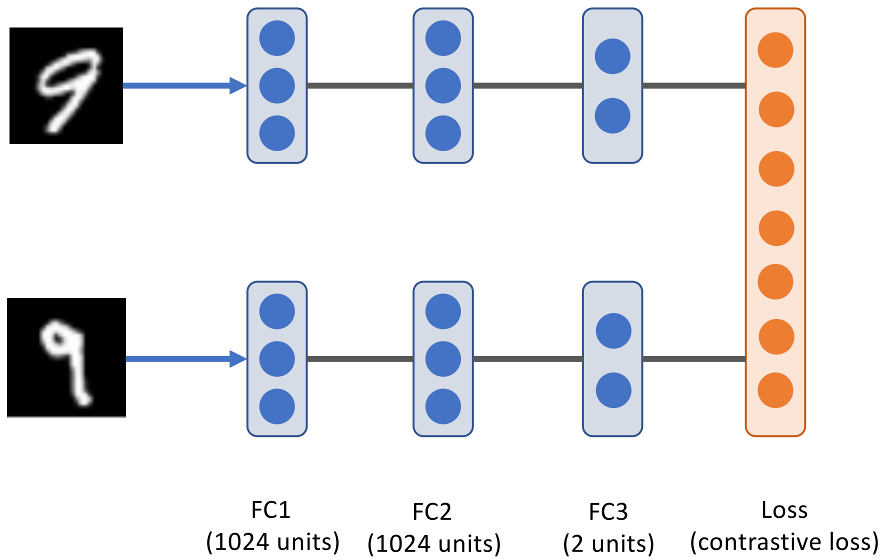
Sentence completion, tweet auto-response

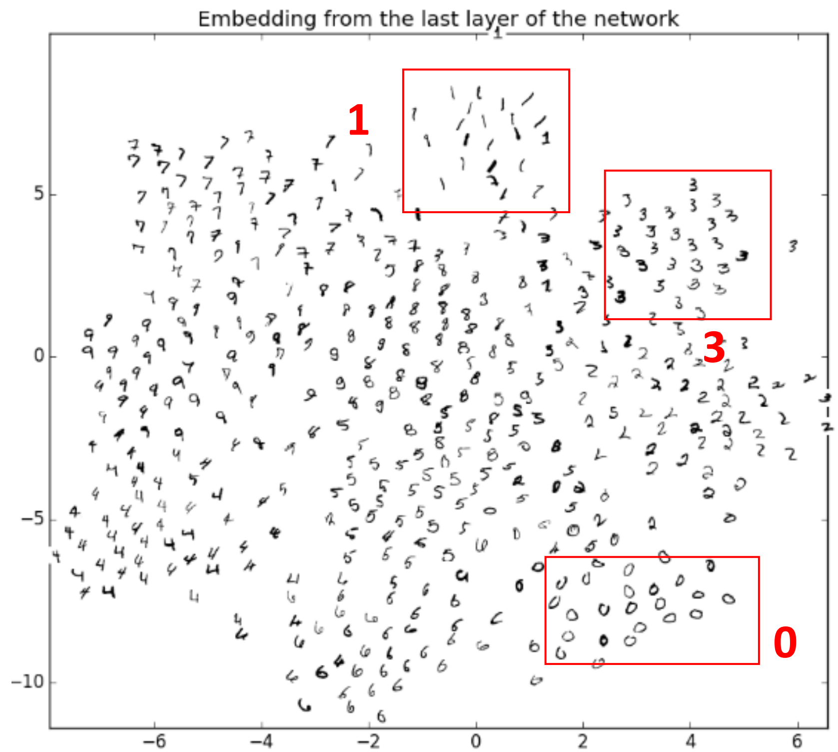
Example:
x : Damn, I have to work overtime this weekend!
y^+: Try to have some rest buddy.
y^-: It is hard to find a job, better start polishing your resume.



Hu, Baotian, et al., Convolutional neural network architectures for matching natural language sentences, NIPS 2014

MNIST Digit Similarity Assessment





Code: @ywpkwon

Summary

- Quantifying “similarity” is essential for data analysis.
- Deep Learning approaches (e.g., Siamese network)
- Many architecture variants for a variety of tasks

References

- Bell, Sean, and Kavita Bala, Learning visual similarity for product design with convolutional neural networks, ACM Transactions on Graphics (TOG), 2015
- Chopra, Sumit, Raia Hadsell, and Yann LeCun, Learning a similarity metric discriminatively, with application to face verification, CVPR 2005
- Zagoruyko, Sergey, and Nikos Komodakis, Learning to compare image patches via convolutional neural networks, CVPR 2015
- Hoffer, Elad, and Nir Ailon, Deep metric learning using triplet network, arXiv:1412.6622
- Simo-Serra, Edgar, et al., Discriminative Learning of Deep Convolutional Feature Point Descriptors, ICCV 2015
- Vo, Nam N., and James Hays, Localizing and Orienting Street Views Using Overhead Imagery, ECCV 2016
- Ahmed, Ejaz, Michael Jones, and Tim K. Marks, An Improved Deep Learning Architecture for Person Re-Identification, CVPR 2015
- Hu, Baotian, et al., Convolutional neural network architectures for matching natural language sentences, NIPS 2014
- Kulis, Brian, Metric learning: A survey, Foundations and Trends in Machine Learning, 2013
- Su, Hang, et al., Multi-view convolutional neural networks for 3d shape recognition, ICCV 2015
- Zheng, Yi, et al., Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks, WAIM 2014
- Yi, Kwang Moo, et al., LIFT: Learned Invariant Feature Transform, arXiv:1603.09114
- Stricker, M.A. and Orengo, M. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology* (pp. 381-392), 1995.