Regression in Deep Learning: Siamese and Triplet Networks

Tu Bui,
John Collomosse

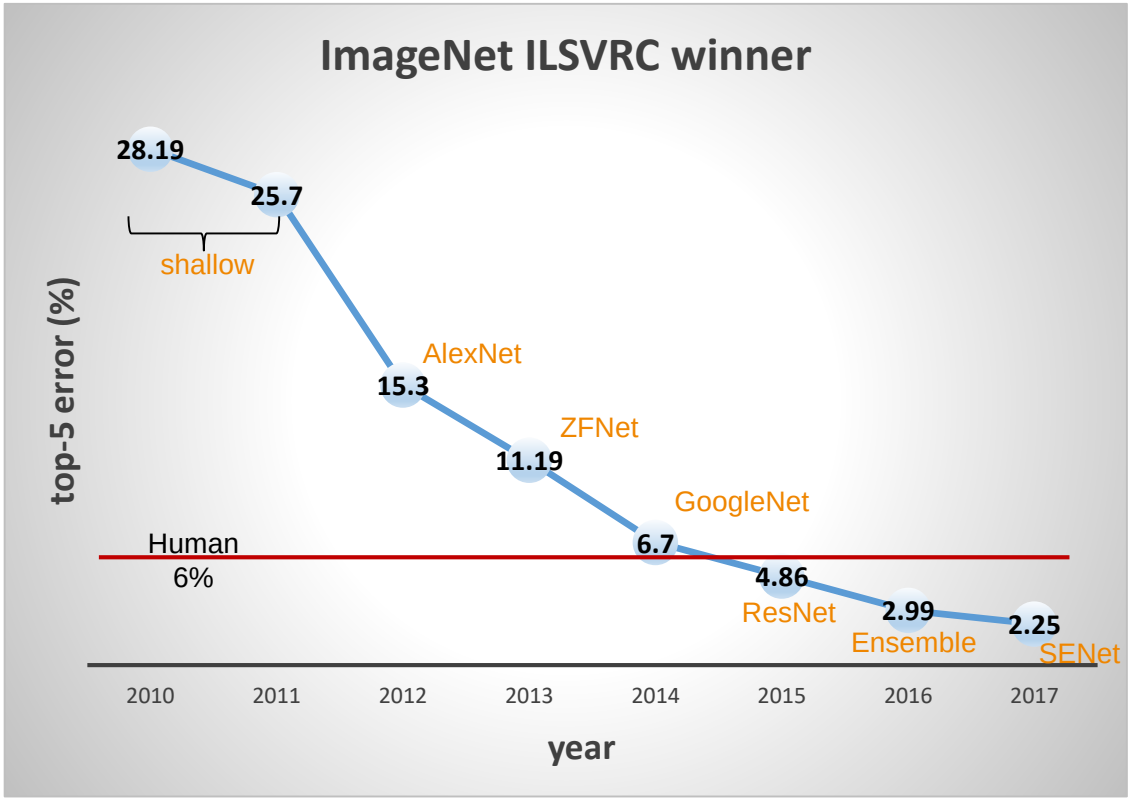
Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, United Kingdom

Leonardo Ribeiro,
Tiago Nazare, Moacir Ponti

Institute of Mathematics and Computer Sciences (ICMC)
University of Sao Paulo, Brazil

Content

Revolution of deep learning in classification

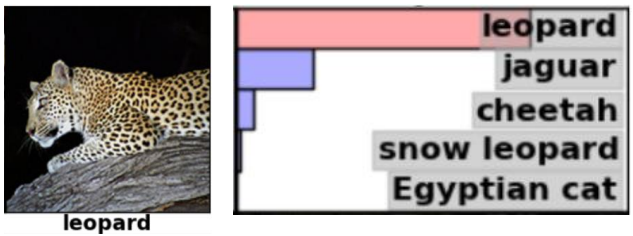


Lower is better

Classification vs. Regression

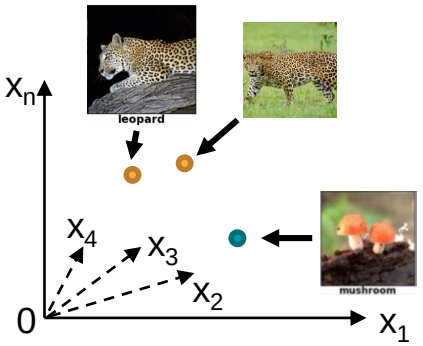
Classification

- **Discrete** set of outputs
- Output: label/class/category



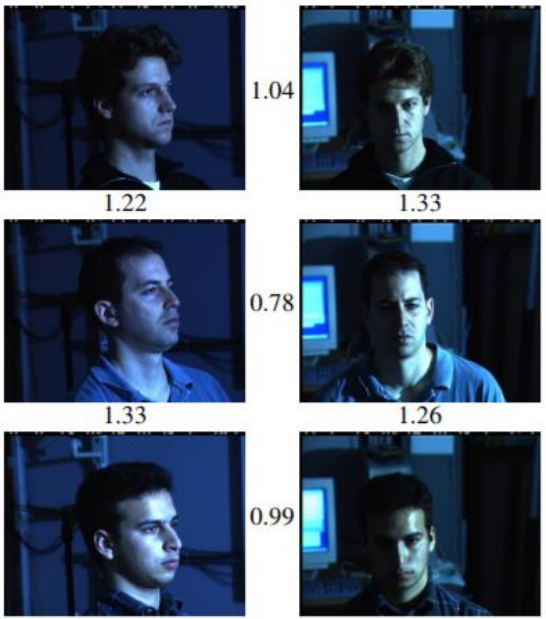
Regression

- **“Continuous”** valued output
- Output: embedding feature



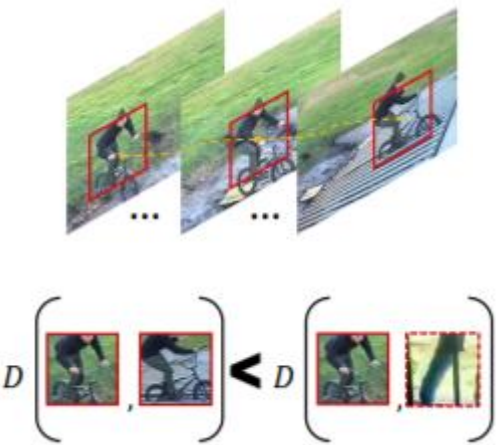
Regression example: intra-domain learning

Face identification



Schroff et al. CVPR 2015

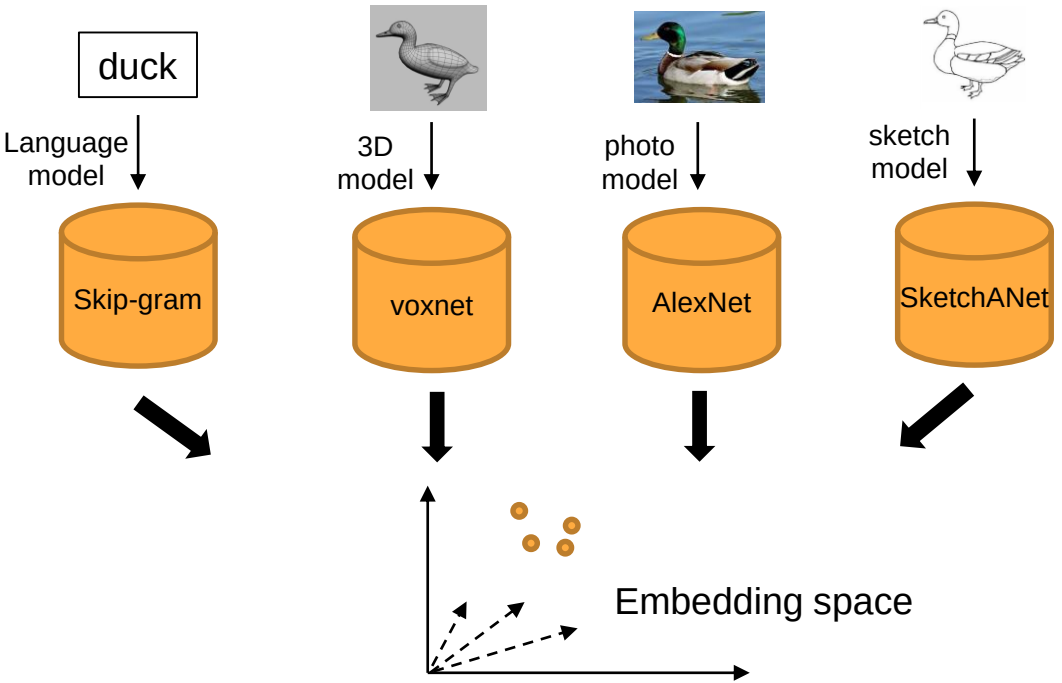
Tracking



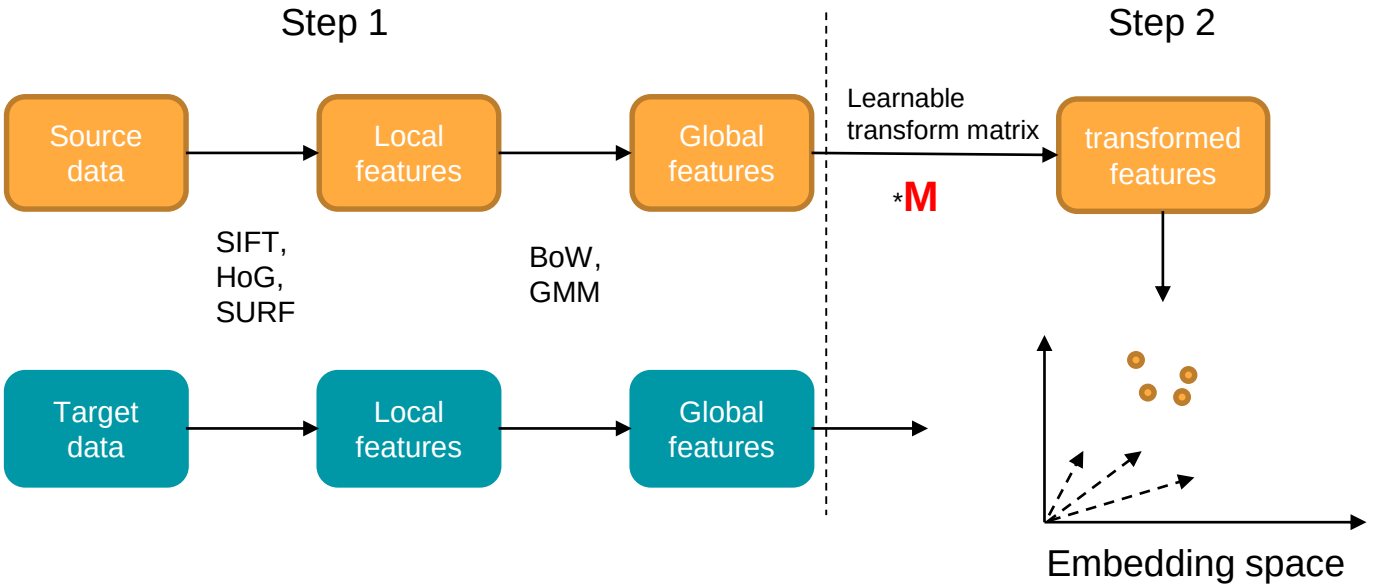
Wang & Gupta ICCV 2015

Regression example: cross-domain learning

Multi-modality visual search



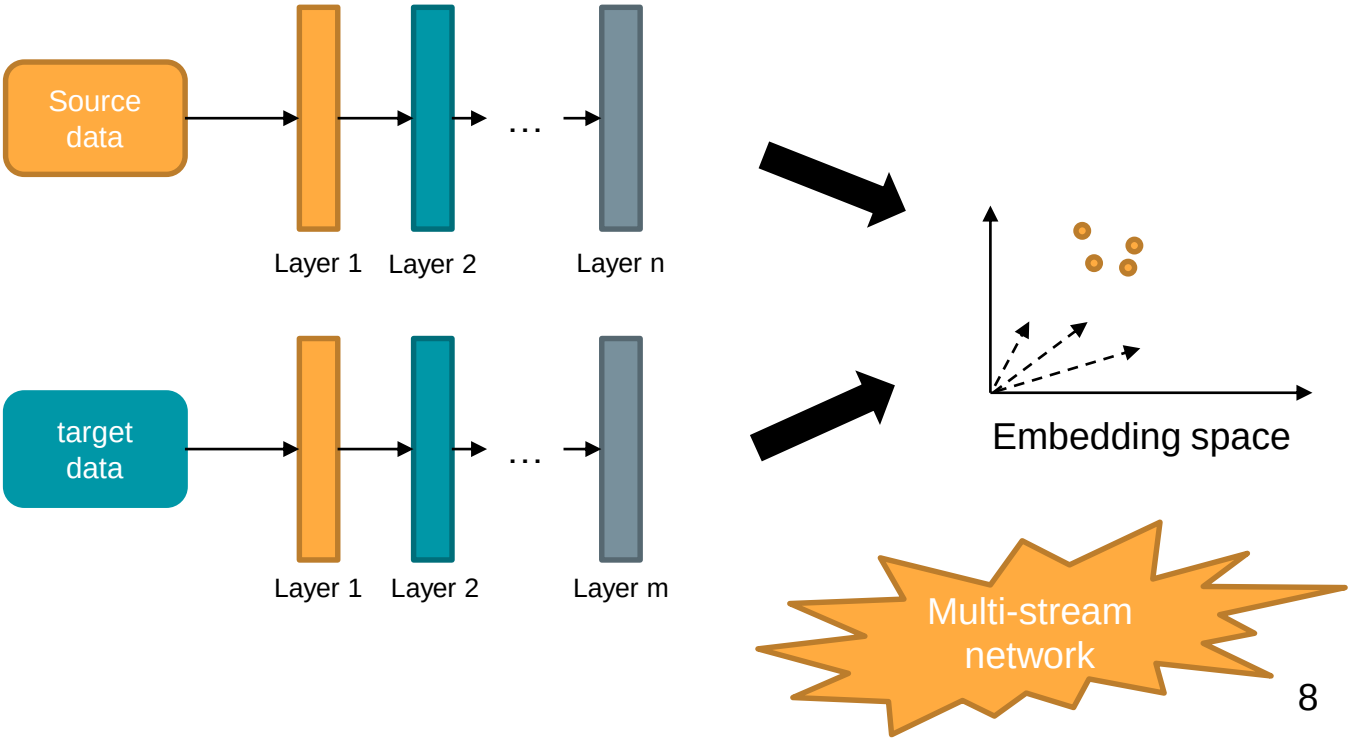
Conventional methods for cross-domain regression



Problem: assume linear transformation between two domains.

End-to-end regression with deep learning

End-to-end learning

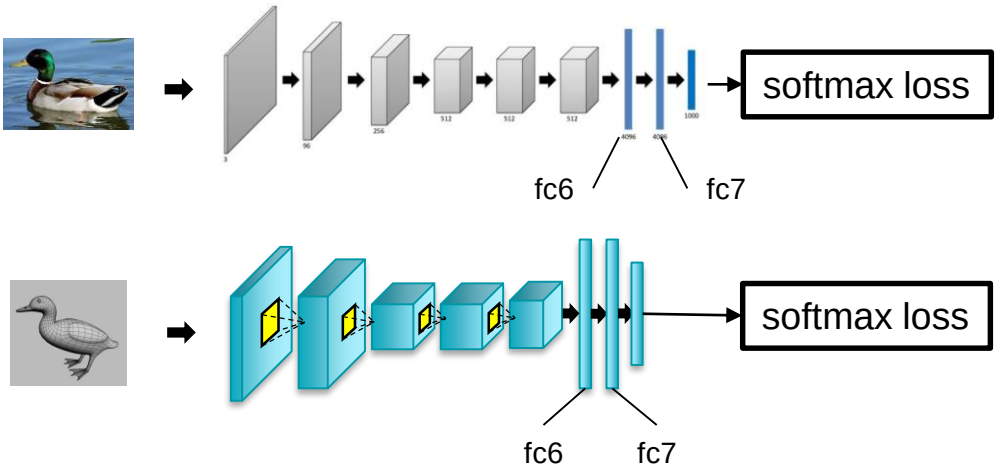


End-to-end regression with multi-stream networks

- Open questions:
  - Network designs?
  - Loss function to be used?

Using output of classification model as feature?

- Not intuitive: different objective function
- Cross-domain learning: training a classification network for each domain separately does not guarantee a common embedding.



Content

The regression problem

Siamese network and contrastive loss

Triplet network and triplet loss

Training tricks

Regression application: sketch-based image retrieval

Limitations and future work

Siamese network and contrastive loss

- Siamese (2-branch) network
- Given an input training pair (x_1,x_2):
  - Label:

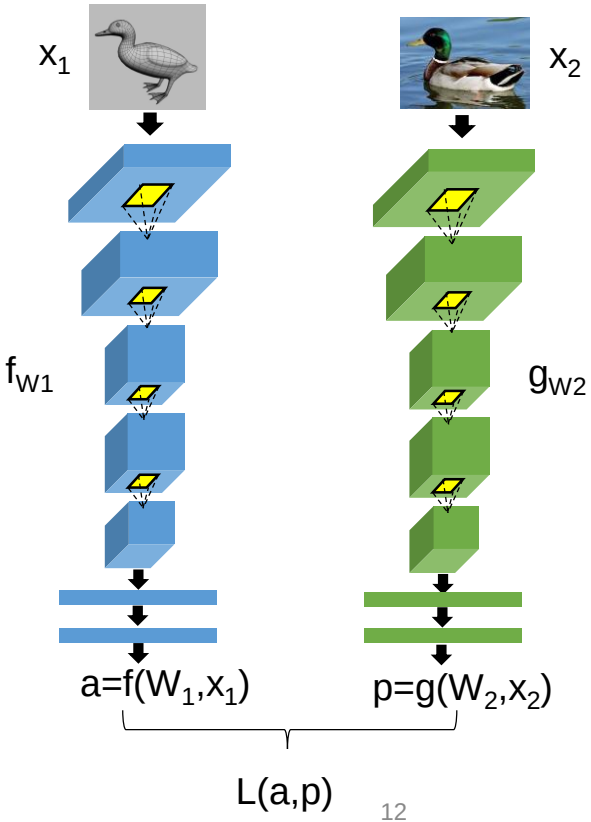
$$y = \begin{cases} 0 & \text{if } (x_1, x_2) \text{ similar pair} \\ 1 & \text{if } (x_1, x_2) \text{ dissimilar pair} \end{cases}$$

  - Network output:

$$a = f(W_1, x_1)$$
$$p = g(W_2, x_2)$$

  - Euclidean distance between outputs:

$$D(W_1, W_2, x_1, x_2) = |a - p|_2 = |f(W_1, x_1) - g(W_2, x_2)|_2$$



Siamese network and contrastive loss

- Contrastive loss equation:

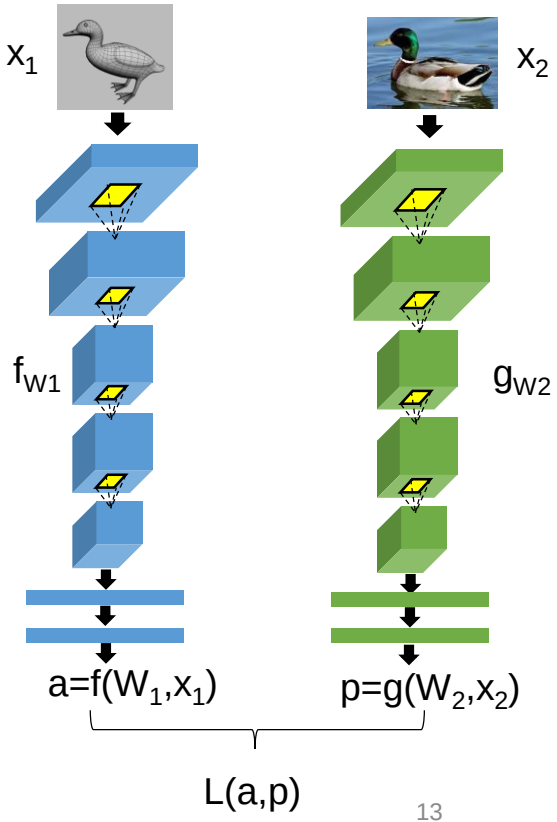
$$\mathcal{L}(W_1, W_2, x_1, x_2) = \frac{1}{2}(1-y)D^2 + \frac{1}{2}y\max\{0, m - D\}^2$$

$$D = |a - p|_2 = |f(W_1, x_1) - g(W_2, x_2)|_2$$

$$y = \begin{cases} 0 & \text{if } (x_1, x_2) \text{ similar pair} \\ 1 & \text{if } (x_1, x_2) \text{ dissimilar pair} \end{cases}$$

margin m: desirable distance for dissimilar pair (x_1,x_2)

- Training: $\underset{W_1, W_2}{\mathbf{argmin}}\, \mathcal{L}$

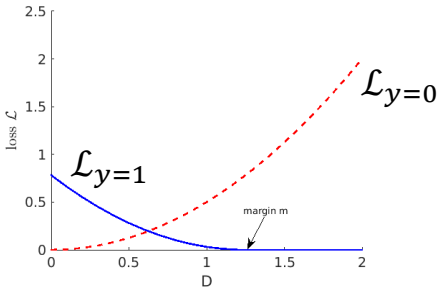


Siamese network and contrastive loss

Contrastive loss functions:

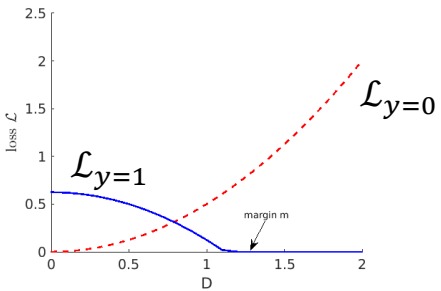
- Standard form*

$$\mathcal{L}(a,p) = \frac{1}{2}(1-y)D^2 + \frac{1}{2}\, y\, \{\max(0, m - D)\}^2$$



- Alternative form**

$$\mathcal{L}(a,p) = \frac{1}{2}(1-y)D^2 + \frac{1}{2}\, y\, \{\max(0, m - D^2)\}$$



*Hadsell et al. CVPR 2006
**Chopra et al. CVPR2005

Content

The regression problem

Siamese network and contrastive loss

Triplet network and triplet loss

Training tricks

Regression application: sketch-based image retrieval

Limitations and future work

Triplet network and triplet loss

- Triplet (3-branch) network

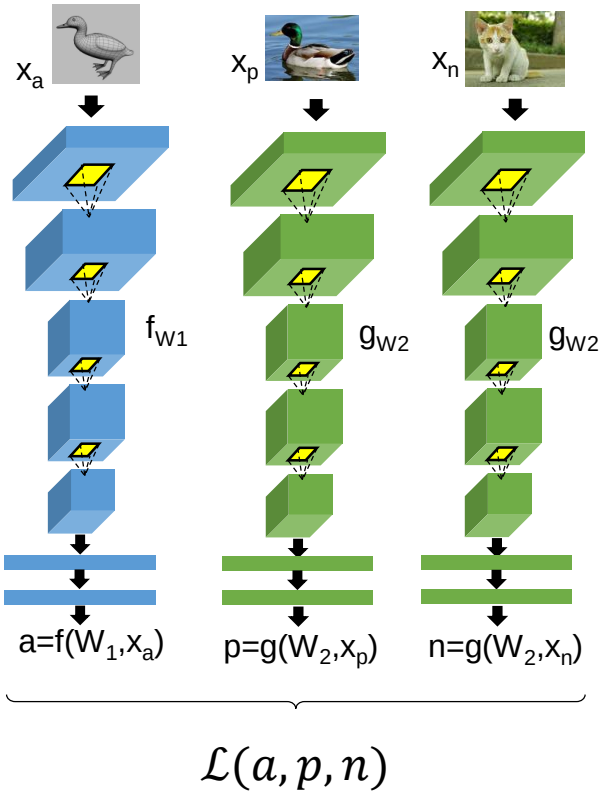
  - Given a training triplet (x_a,x_p,x_n): x_a – anchor; x_p – positive (similar to x_a); x_n – negative (dissimilar to x_a)

  - Pos/neg branches always share weights.

  - Anchor branch can share weights (intra-domain learning) or not (cross-domain learning).

  - Network outputs:

$$a = f(W_1, x_a)$$
$$p = g(W_2, x_p)$$
$$n = g(W_2, x_n)$$



Triplet network and triplet loss

Triplet loss equation:

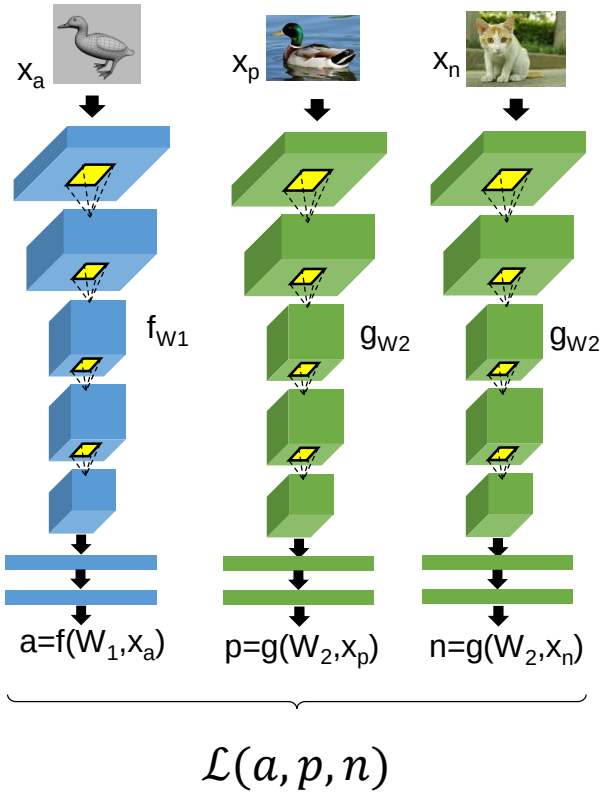
$$L(a,p,n) = \frac{1}{2}\{\max(0, m + D^2(a,p) - D^2(a,n)\}$$

- Standard form*:

$$D(u,v) = |u - v|_2$$

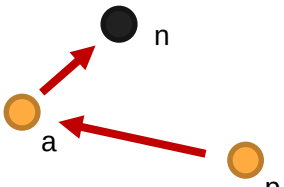
- Alternative form**:

$$D(u,v) = \sqrt{1 - \frac{u.v}{|u|_2|v|_2}}$$

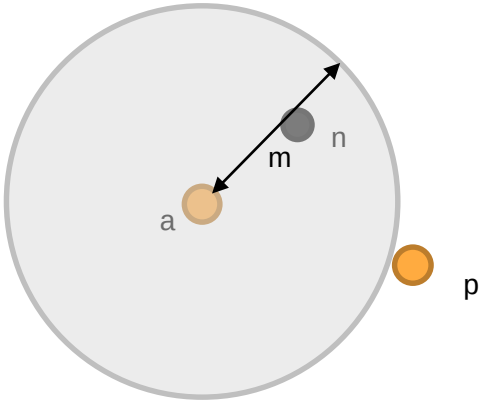


*Schroff et al. CVPR 2015
**Wang et al. ICCV 2015

Siamese vs. Triplet

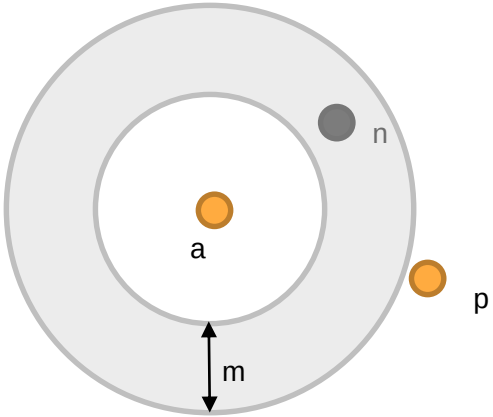


Before training



Contrastive loss

$$L(a,p) = \frac{1}{2}(1-y)|a-p|_2^2 + \\ + \frac{1}{2}\, y\, \{\max(0, m - |a-p|_2^2\}$$



Triplet loss

$$L(a,p,n) = \frac{1}{2}\ \{\max(0, m + \\ + |a-p|_2^2 - |a-n|_2^2\}$$

Siamese or triplet?

Depending on data, training strategies, network design and more:

- Siamese superior
  - Radenovie et al. ECCV 2016

- Triplet superior
  - Hoffer & Ailon. SBPR 2015.
  - Bui et al. arxiv 2016.

Content

The regression problem

Siamese network and contrastive loss

Triplet network and triplet loss

Training tricks

Regression application: sketch-based image retrieval

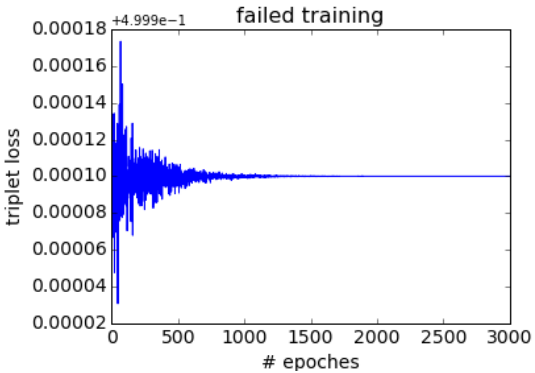
Limitations and future work

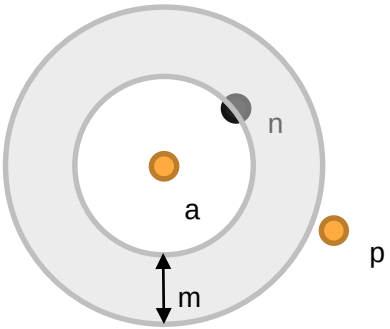
Training trick #1: solving gradient collapsing problem

- The gradient collapsing problem

$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\{\max(0, \mathrm{m} + |a_i - p_i|_2^2 - |a_i - n_i|_2^2)\}$$

Margin m = 1.0





expected

reality

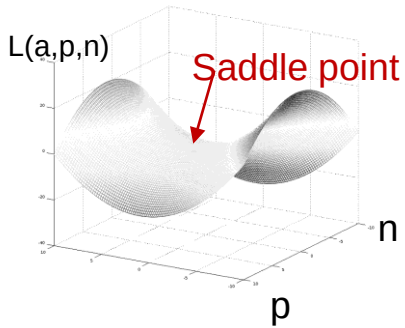
Training tricks #1

- Solution for gradient collapsing:

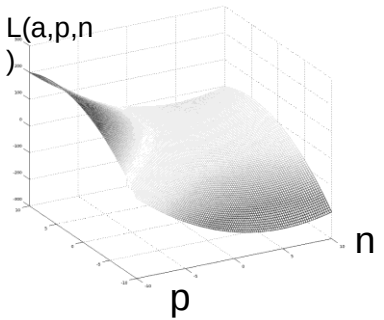
○Combine regression and classification loss for better regularisation.

○Change loss function.

$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\{\max(0, \mathrm{m} + |ka_i - p_i|_2^2 - |ka_i - n_i|_2^2)\}$$



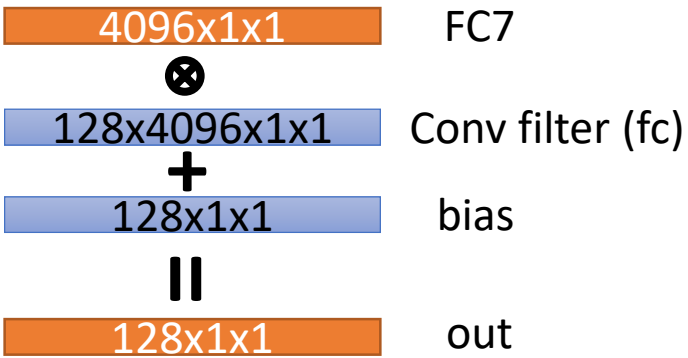
$a \approx p, n$

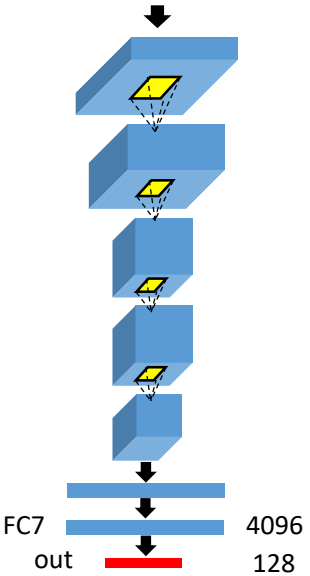


$a \neq p, n$

Training tricks #2: dimensional reduction

- Conventional methods:
  - Redundant analysis on a fixed set of features.
  - E.g. Principal Component Analysis (PCA), Product quantisation, etc
- Dimensional reduction in CNN: part of the training process

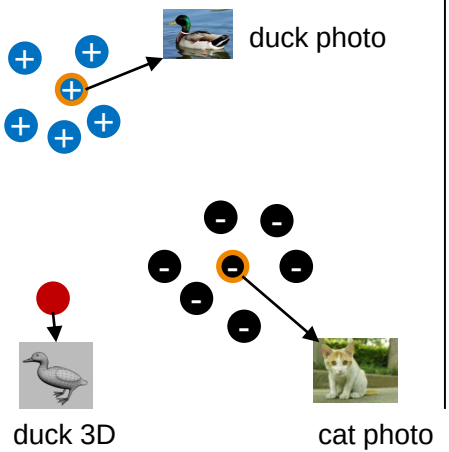




Training tricks #3: hard negative mining

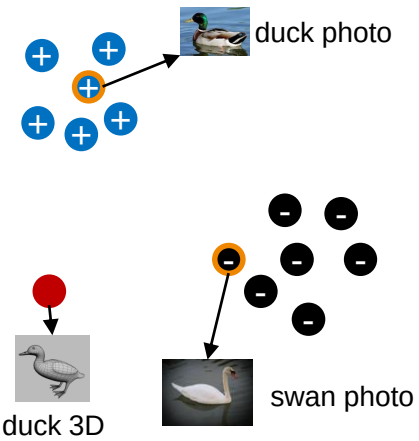
Random paring

Positive and negative samples are selected randomly.



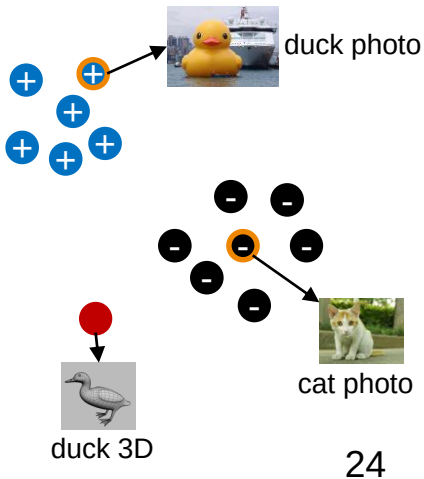
Hard negative mining

Negative example is the nearest irrelevant neighbor to the anchor.



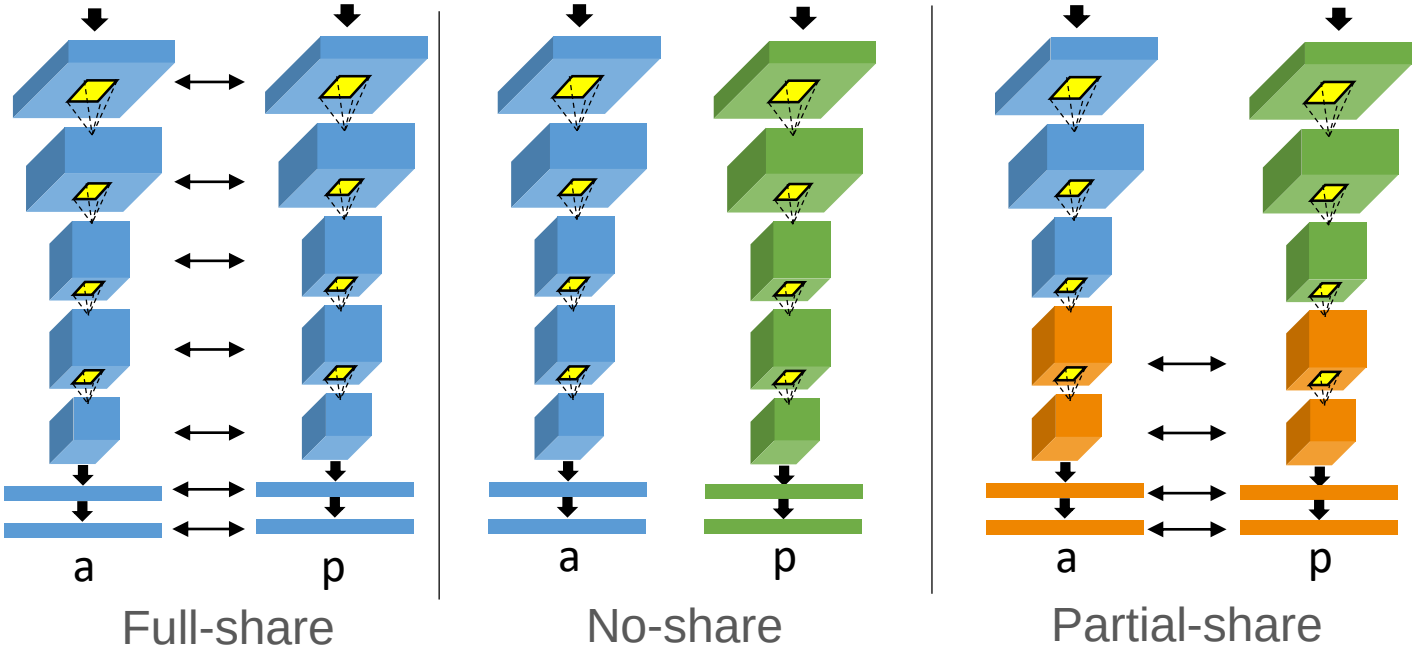
Hard positive mining

Positive example is the farthest relevant neighbor to the anchor.



Training tricks #4: layer sharing

- Consider sharing the anchor with the pos/neg branches



Other training tricks

- Data augmentation:
  - Random crop, rotation, scaling, flip, whitening…
- Dropout:
  - Randomly disable neurons
- Regularisation:
  - Add parameter magnitude to loss
  - $\mathcal{L}_{total}(W, X) = \mathcal{L}_{contrastive,triplet}(W, X) + |W|^2$

Content

The regression problem

Siamese network and contrastive loss

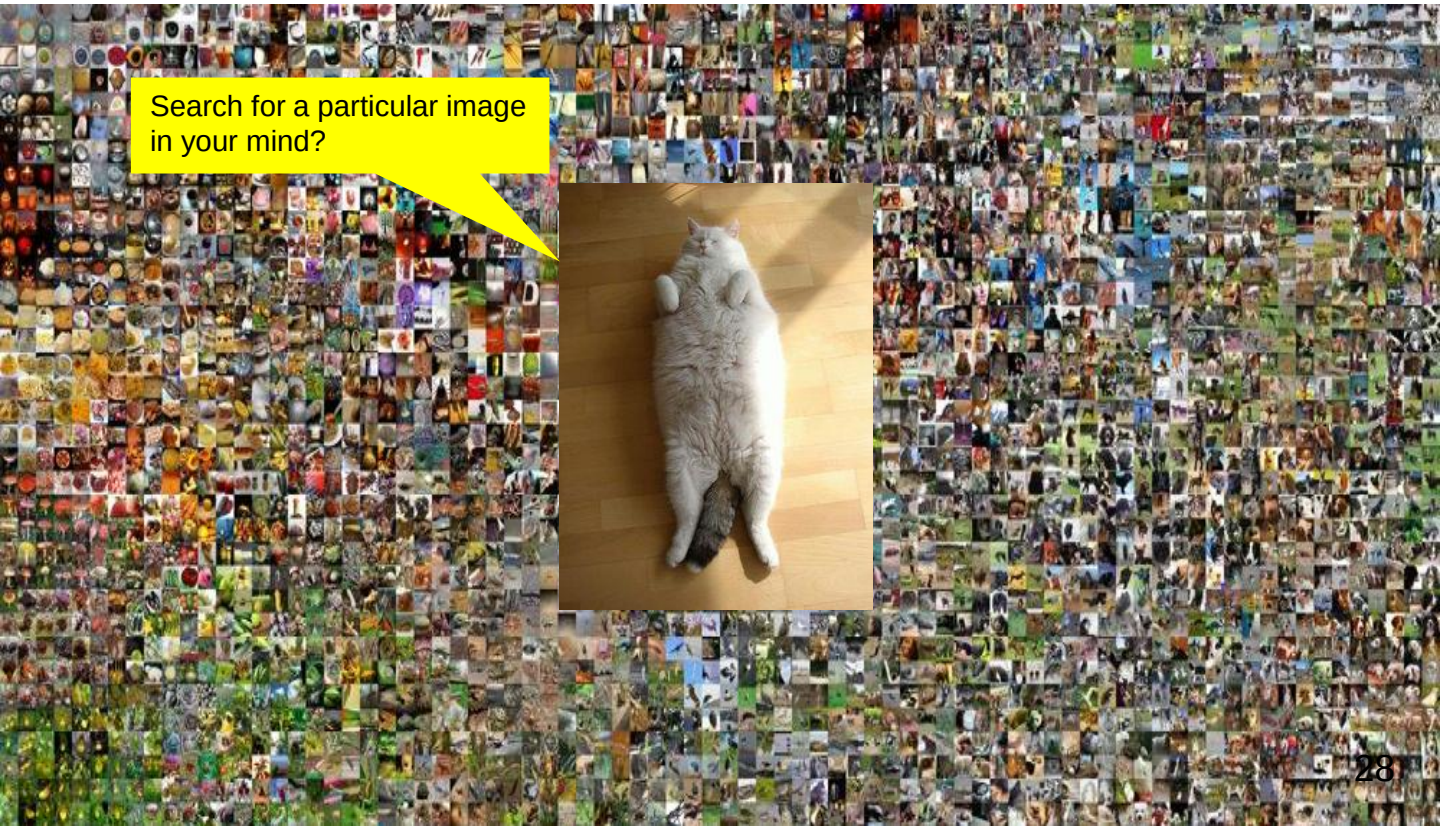
Triplet network and triplet loss

Training tricks

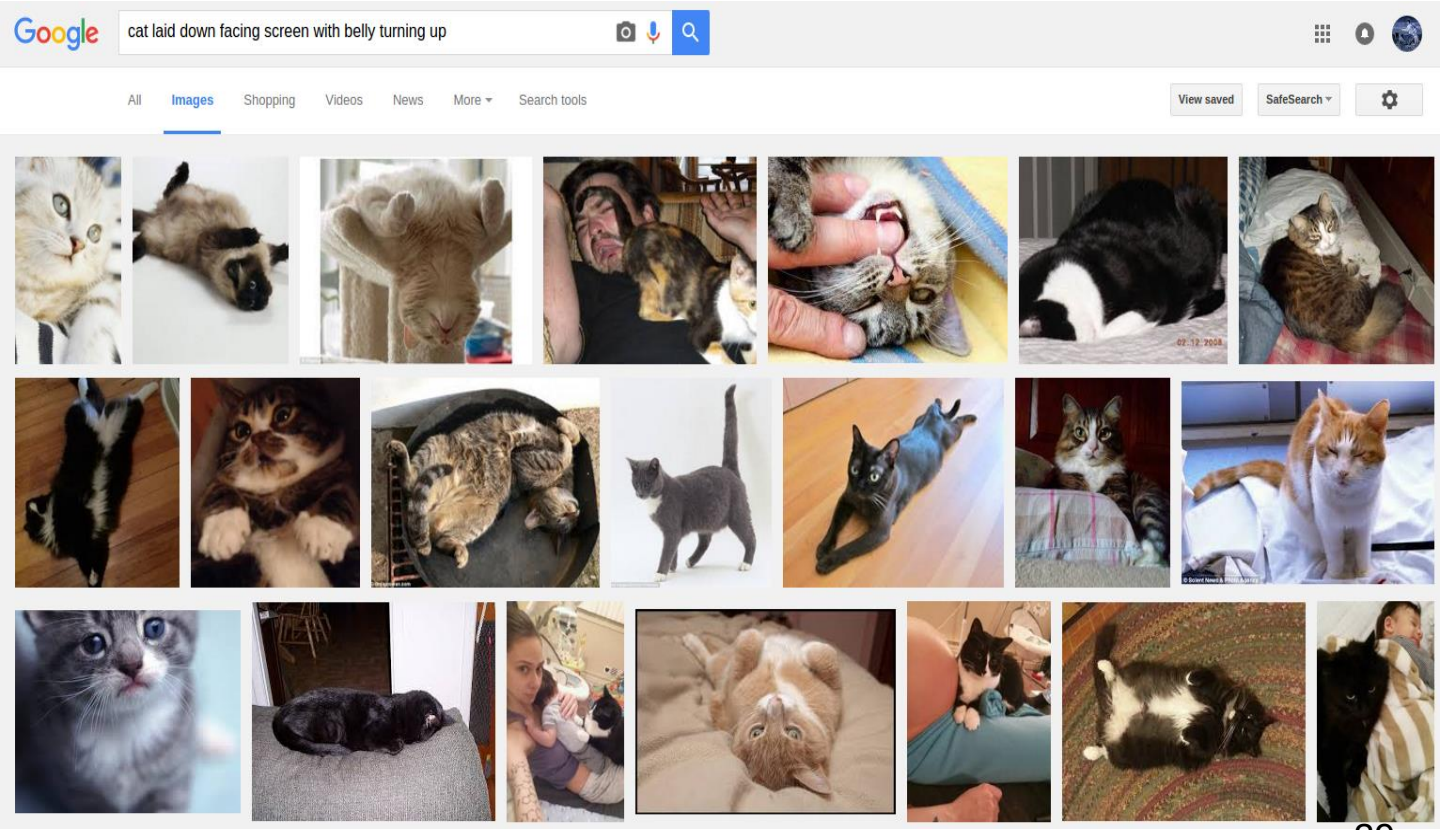
Regression application: sketch-based image retrieval

Limitations and future work

Regression application: sketch-based image retrieval (SBIR)



Text search?


Google
cat laid down facing screen with belly turning up
All
Images
Shopping
Videos
News
More
Search tools
View saved
SafeSearch

Sketch-based Image Retrieval (SBIR)

sketch

retrieval

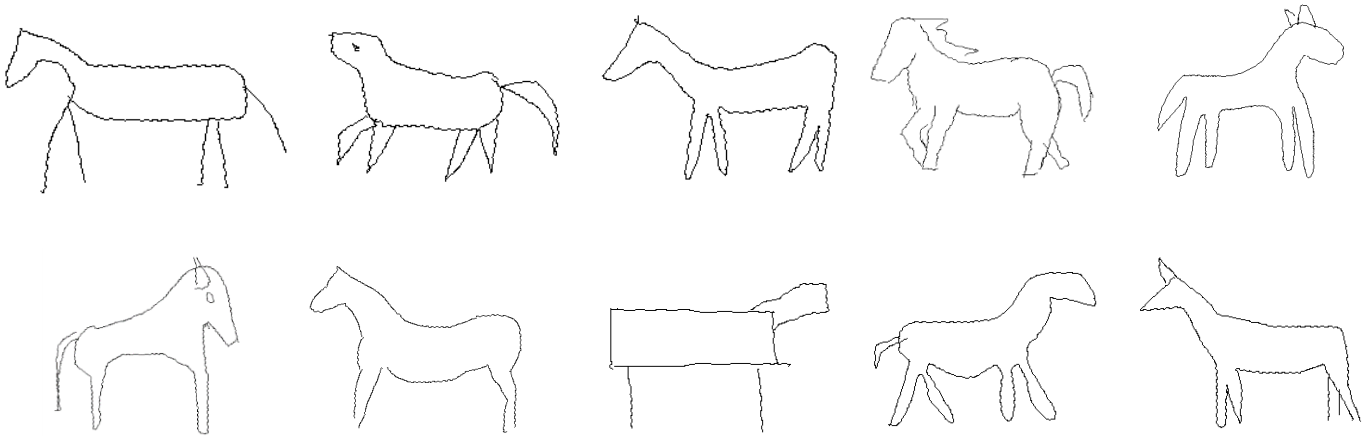
Existing applications

Google Emoji Search

Detexify: latex symbol search

Challenges

- Free-hand sketch is usually messy.



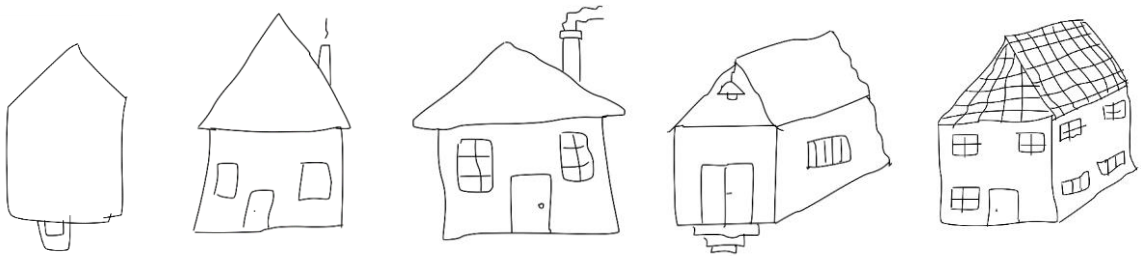
Horse category

Flickr-330 dataset, Hu et al. 2013

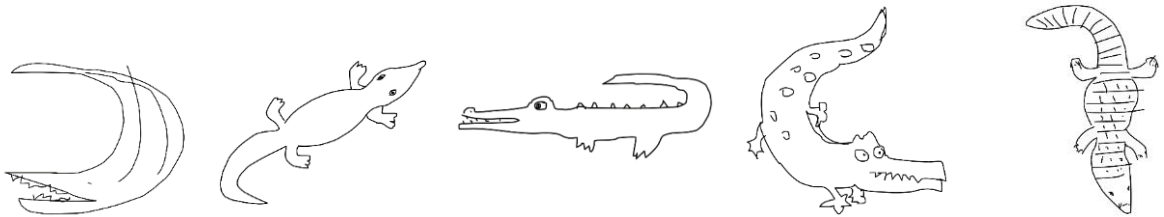
Challenges

- Various levels of abstraction.

House



Crocodile

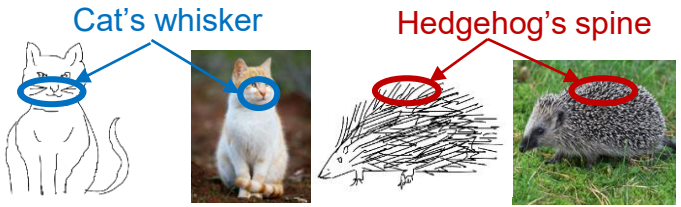


TU-Berlin dataset, Eitz et al. 2012

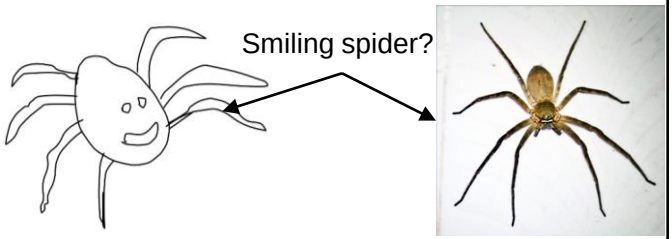
Challenges

- Domain gap: sketch does not always describe real-life object accurately.

Caricature



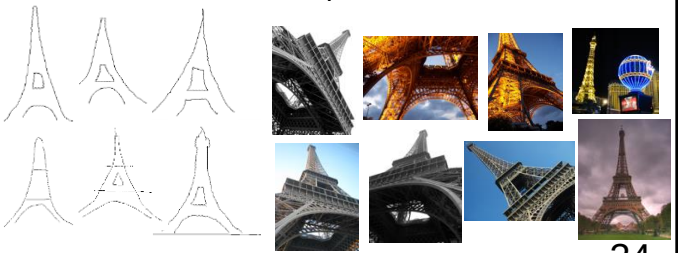
Anthropomorphism



Simplification

Category
"person walking"
TU-Berlin

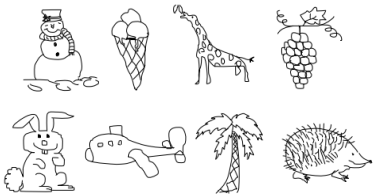
Viewpoint



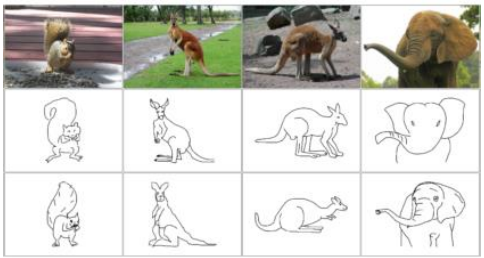
Challenges

- Limited #sketch datasets.
  - **Flickr15K**: 330 sketches + 15k photos @33 classes
  - **TU-Berlin**: 20k sketches@250 classes
  - **Sketchy**: ~75k sketches + 12.5k photos @125 classes
  - **New Google Quickdraw**: 50M sketches @345 classes

Flickr15K [Hu et al. 2013]



TU-Berlin [Eitz et al. 2012]



Sketchy [Sangkloy et al. 2016]

SBIR evaluation metric

- Evaluation metric
  - **Mean Average Precision (mAP)**
  - Precision-recal (PR) curve
  - Kendal rank correlation coefficient

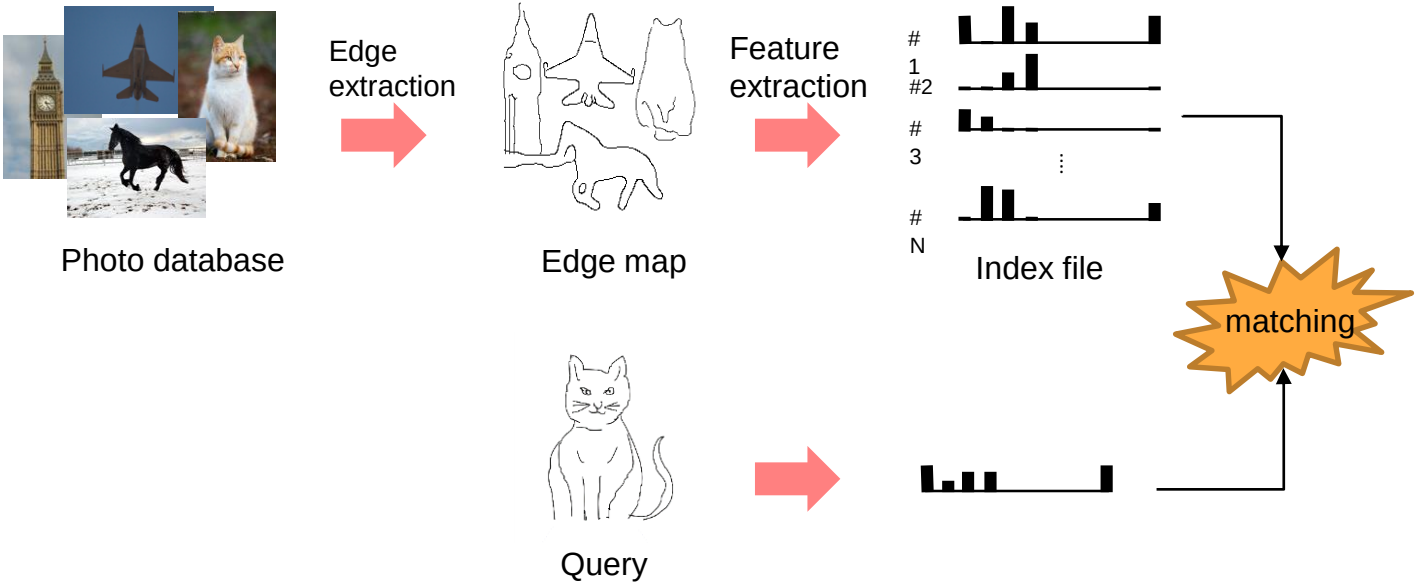
$$P(k) = \frac{\#\ relevant\ in\ top\ k\ results}{k}$$

$$AP = \frac{\sum_{k=1}^{N} P(k) \times rel(k)}{\#\ relevant\ images}$$

$$mAP = \frac{1}{Q} \times \sum_{q \in Q} AP_q$$

Background

- Conventional shallow SBIR framework

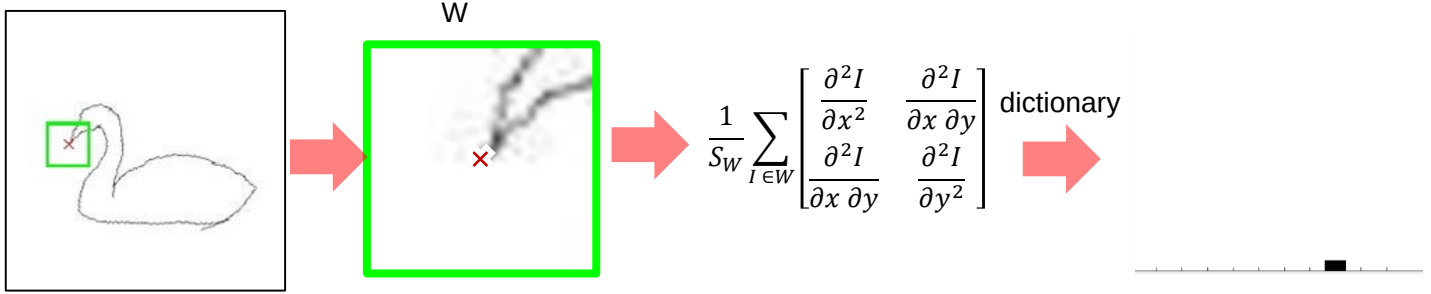


Background: hand-crafted features

- Structure tensor [Eitz,2010]

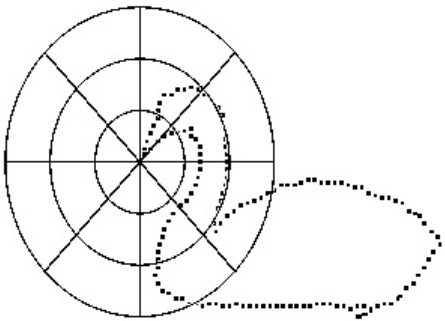
Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98



Background: hand-crafted features

- Shape context [Mori, 2005]



Flickr15K benchmark

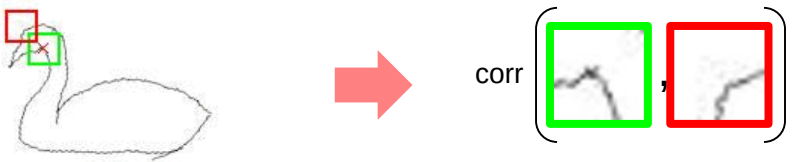
Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	**8.14**

Background: hand-crafted features

- Self similarity [Shechtman, 2007]

Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	**9.57**

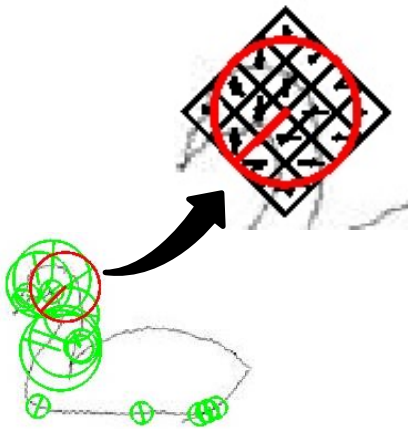


Background: hand-crafted features

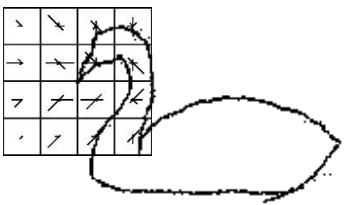
- SIFT [Lowe, 2004]
- HoG [Dalas, 2005]

Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	**9.11**
HoG [Dalas, 2005]	**10.93**



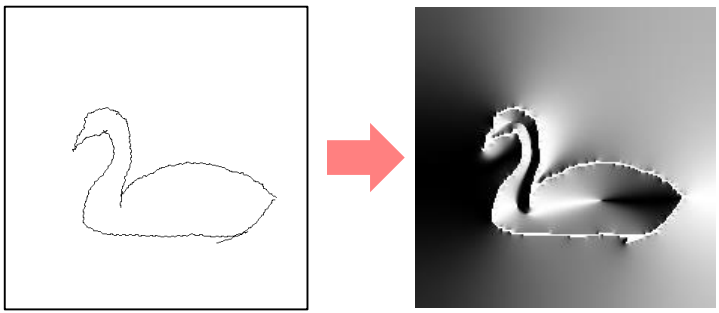
SIFT



HoG

Background: hand-crafted features

- GF-HoG [Hu et al. CVIU 2013]
- Color GF-HoG [Bui et al. ICCV 2015]



Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	9.11
HoG [Dalas, 2005]	10.93
GF-HoG [Hu, 2013]	**12.22**
Color GF-HoG [Bui, 2015]	**18.20**

Background: hand-crafted features

- PerceptualEdge [Qi, 2015]

Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	9.11
HoG [Dalas, 2005]	10.93
GF-HoG [Hu, 2013]	12.22
Color GF-HoG [Bui, 2015]	18.20
PerceptualEdge [Qi, 2015]	**18.37**

Back ground: deep features

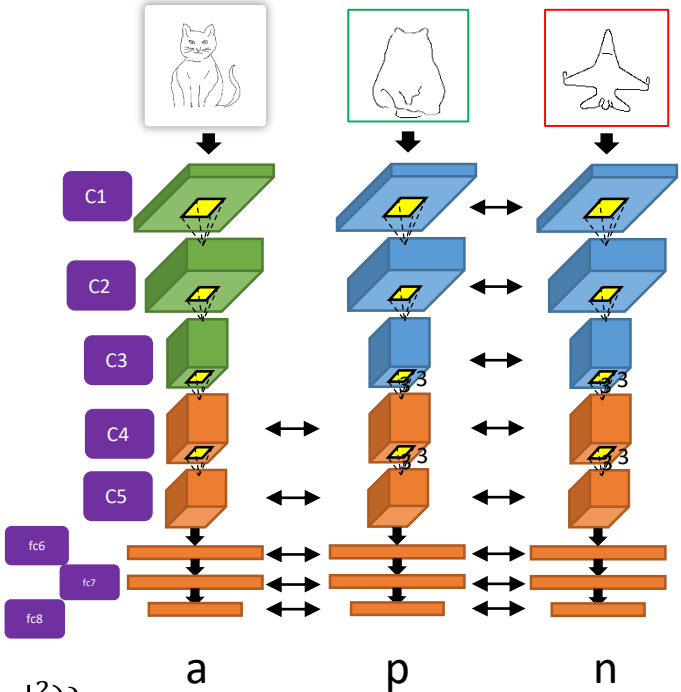
- Siamese network with contrastive loss
- Qi et al. ICIP 2016
  - Sketch-edgemap
  - Fully shared

Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	9.11
HoG [Dalas, 2005]	10.93
GF-HoG [Hu, 2013]	12.22
Color GF-HoG [Bui, 2015]	18.20
PerceptualEdge [Qi, 2015]	18.37
Siamese network [Qi, 2016]	**19.54**

Triplet network for SBIR

- Sketch-edgemap
- CNN architecture: Sketch-A-Net [Yu, 2015]
- Output dimension: 100
- Share layers: Conv 4-5, FC 6-8
- Loss:



$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\{\max(0, \mathrm{m} + |ka_i - p_i|_2^2 - |ka_i - n_i|_2^2)\}$$

k = 2.0

Training procedure

- Images:
  - 25k photos: 100 photos/class.
  - Edge extraction: gPb [Arbelaez, 2011].
  - Mean subtraction, random crop/rotation/scaling/flip.
- Sketches:
  - 20k sketches: 20s training, 60s validation per class.
  - Skeletonisation.
  - Mean subtraction, random crop/rotation/scaling/flip.
  - Random stroke removal.
- Triplet formation:
  - Random selection pos/neg samples.
- Training:
  - 10k epochs. Multistep decreasing learning rate k = 10^{-2} – 10^{-6}.

crop

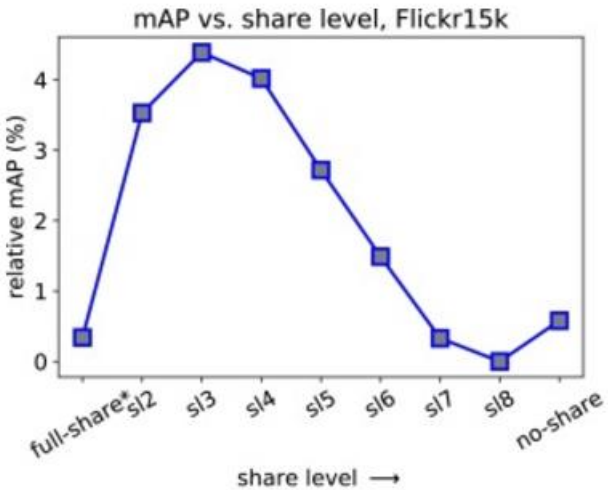
rotation

scaling

flip

Stroke removal

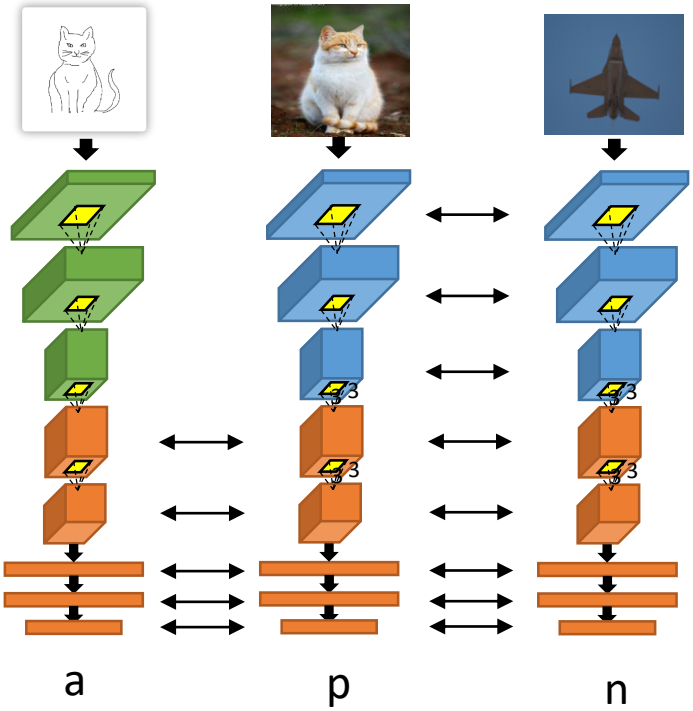
Results

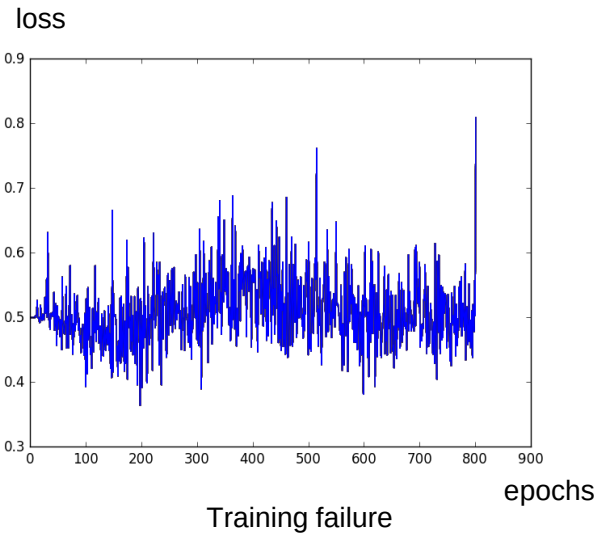


Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	9.11
HoG [Dalas, 2005]	10.93
GF-HoG [Hu, 2013]	12.22
Colour GF-HoG [Bui, 2015]	18.20
PerceptualEdge [Qi, 2015]	18.37
Single CNN	18.76
Siamese network [Qi, 2016]	19.54
Triplet full-share [Bui, 2016]	**20.29**
Triplet no-share [Bui, 2016]	**20.93**
Triplet half-share [Bui, 2016]	**24.45**

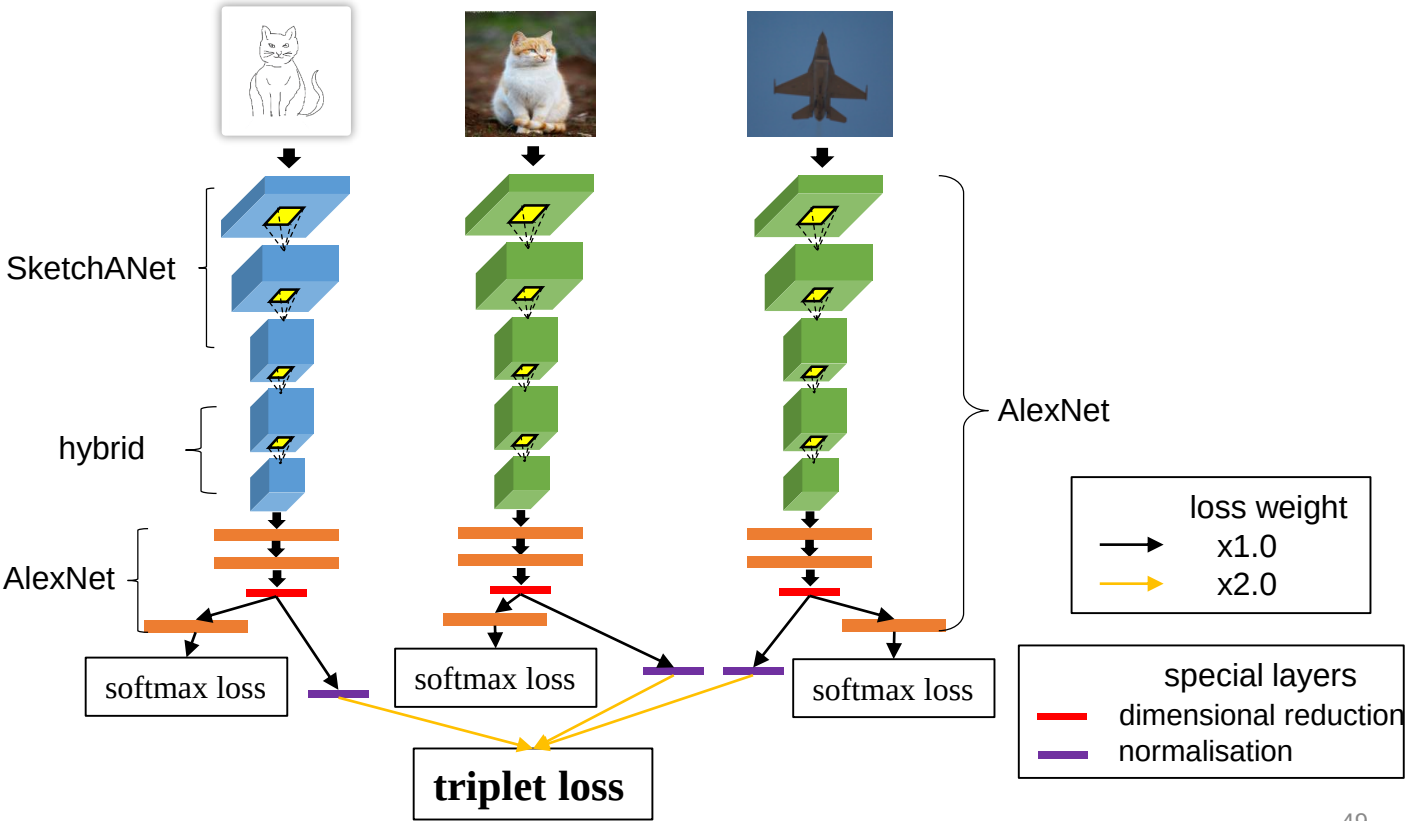
Sketch-photo direct matching





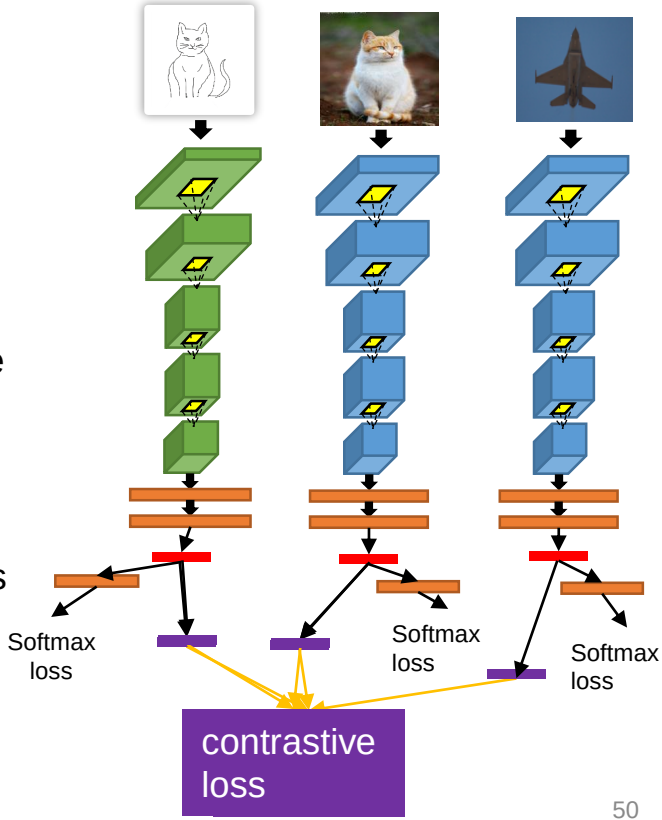
Training failure

Sketch-photo direct matching



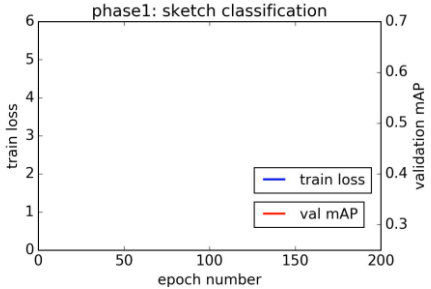
Multi-stage training procedure

- Stage 1: train unshared layers
  - Train Sketch branch from scratch.
  - Finetune image branch from AlexNet
- Stage 2: train shared layers
  - Form a 2-branch network with pretrained weights.
  - Freeze unshared layers.
  - Train the shared layers with contrastive loss + softmax loss.
- Stage 3: regression with triplet loss
  - Form a triplet network.
  - Unfreeze the all layers.
  - Train the whole network with triplet loss + softmax loss.



Training results

Phase 1



Sketch branch

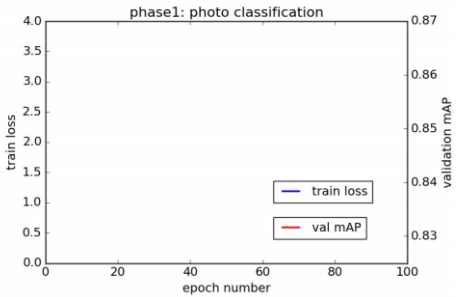
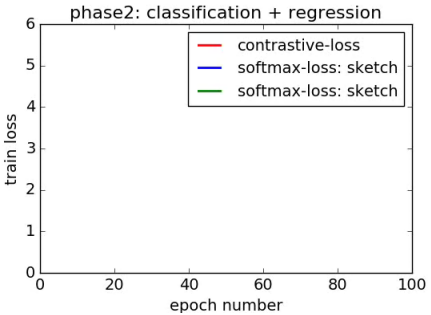
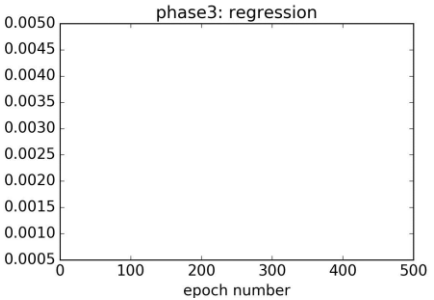


Image branch

Phase 2



Siamese network



Triplet network

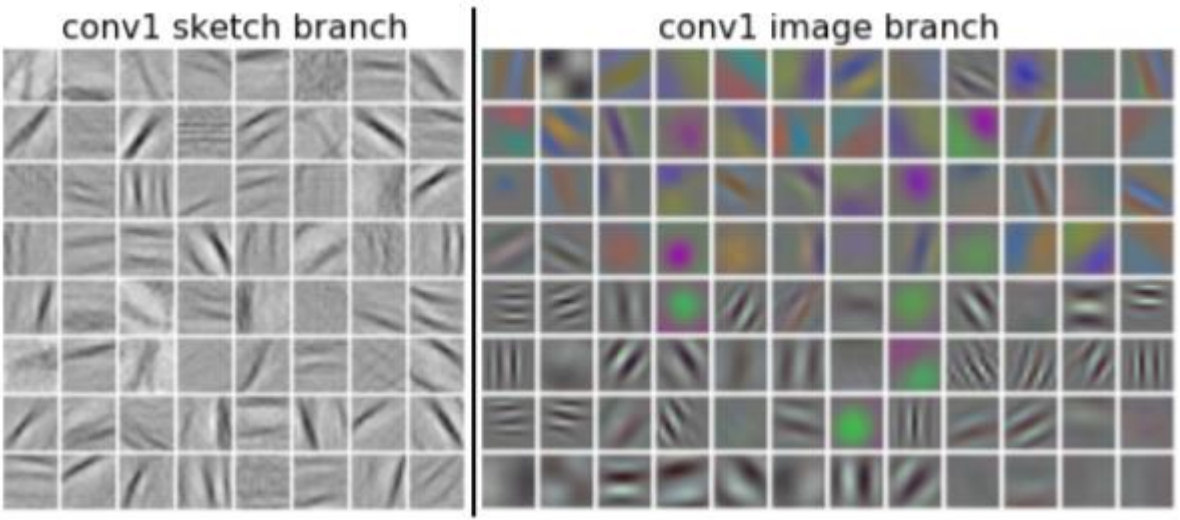
Phase 3

Results

Flickr15K benchmark

Method	mAP(%)
Structure Tensor [Eitz, 2010]	7.98
Shape Context [Mori, 2005]	8.14
SSIM [Shechtman, 2007]	9.57
SIFT [Lowe, 2004]	9.11
HoG [Dalas, 2005]	10.93
GF-HoG [Hu, 2013]	12.22
Colour GF-HoG [Bui, 2015]	18.20
PerceptualEdge [Qi, 2015]	18.37
Single CNN	18.76
Siamese network [Qi, 2016]	19.54
Sketch-edgemap triplet [Bui, 2016]	**24.45**
Sketch-photo triplet	**31.38**

Layer visualisation



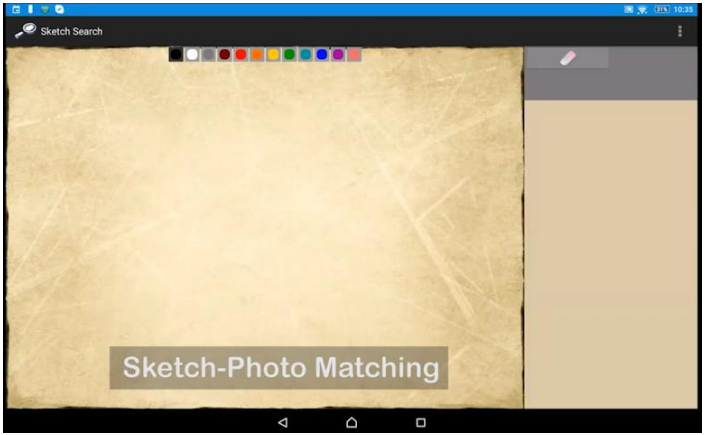
64 15x15 filters in conv1 layer
SketchANet

96 11x11 filters in conv1 layer
AlexNet

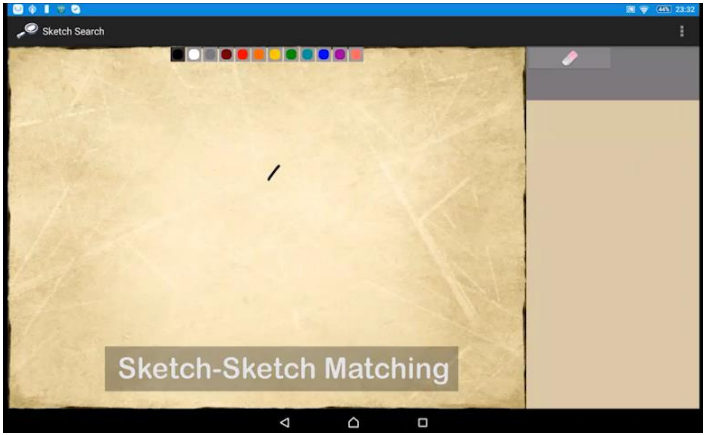
SBIR example

Demo: SketchSearch

Sketch-based Image Retrieval



Sketch Retrieval



Content

The regression problem

Siamese network and contrastive loss

Triplet network and triplet loss

Training tricks

Regression application: sketch-based image retrieval

Limitations and future work

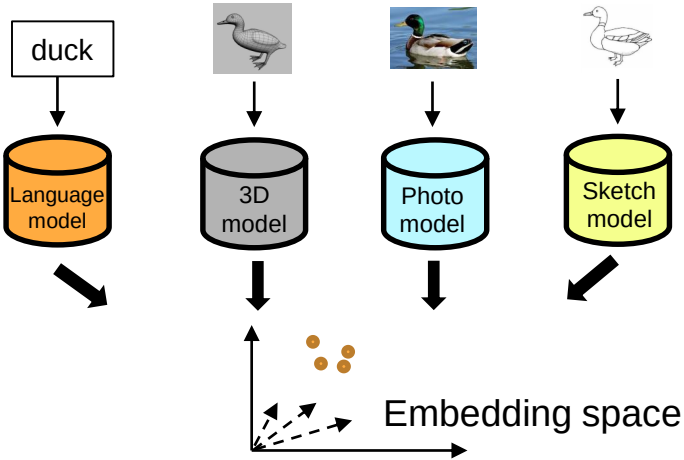
Limitations

- Hard to train a regression model.
- Need labelled datasets.
- Real-life sketch can be very complicated

Guernica
by Pablo Picasso, 1937

Future work

- Multi-domain regression e.g. 3D, text, photo, sketch, depth-map, cartoon…



Castrejon, 2016

Siddiquie, 2014

- Toward unsupervised deep learning:

- Labelled image set, unlabelled or no sketch set

Radenovic, 2017

- Completely unsupervised: Auto-encoder, Generative Adversaries Network (GAN)

Thank you for listening

https://sites.google.com/view/tubui/research