Domain-Adaptive Deep Network Compression

&

Deep Learning from Rankings

UPC January 2018

Joost van de Weijer
Universitat Autònoma de Barcelona
Computer Vision Center Barcelona

Domain-Adaptive Deep Network Compression

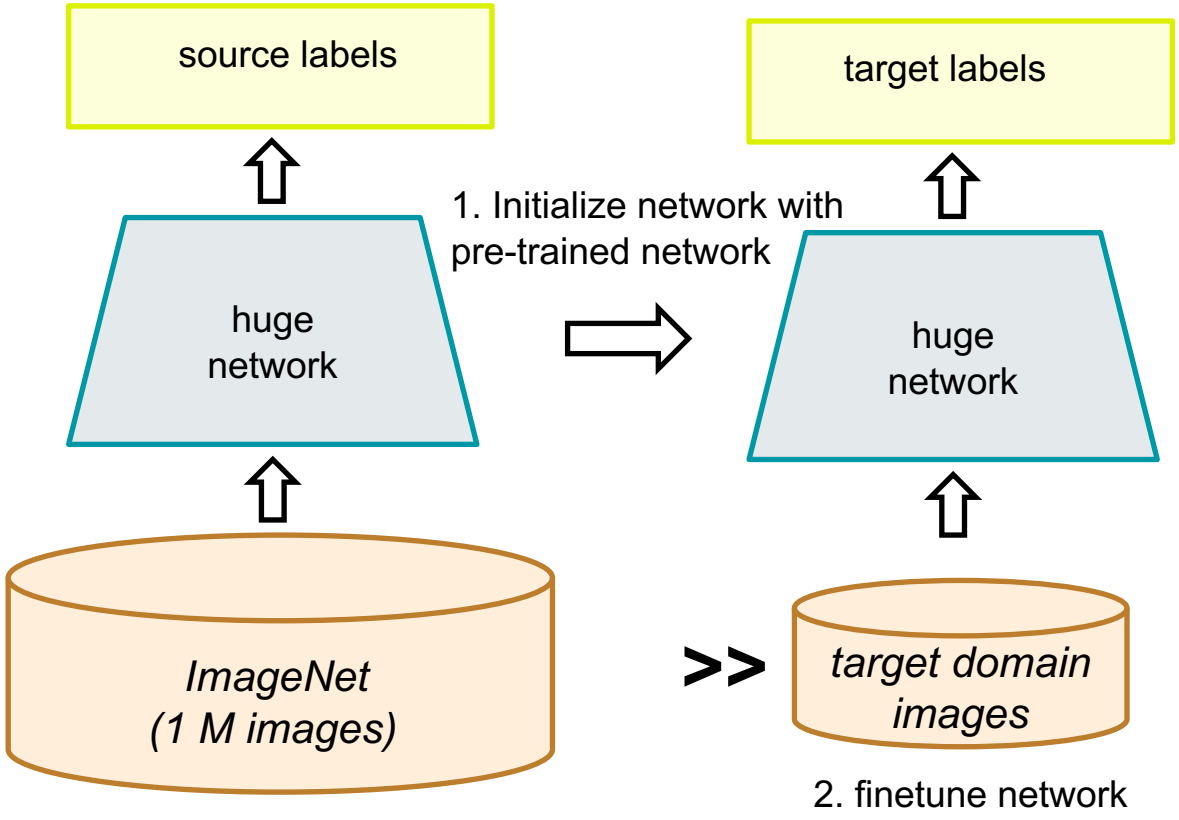
Marc Masana, Joost van de Weijer, Andrew Bagdanov, Jose Álvarez, Luis Herranz

ICCV 2017

LAMP The Learning and Machine Perception Team

CVC Centre de Visió per Computador

Problem Statement



Problem statement:
Do we require the huge network on the small target domain ?

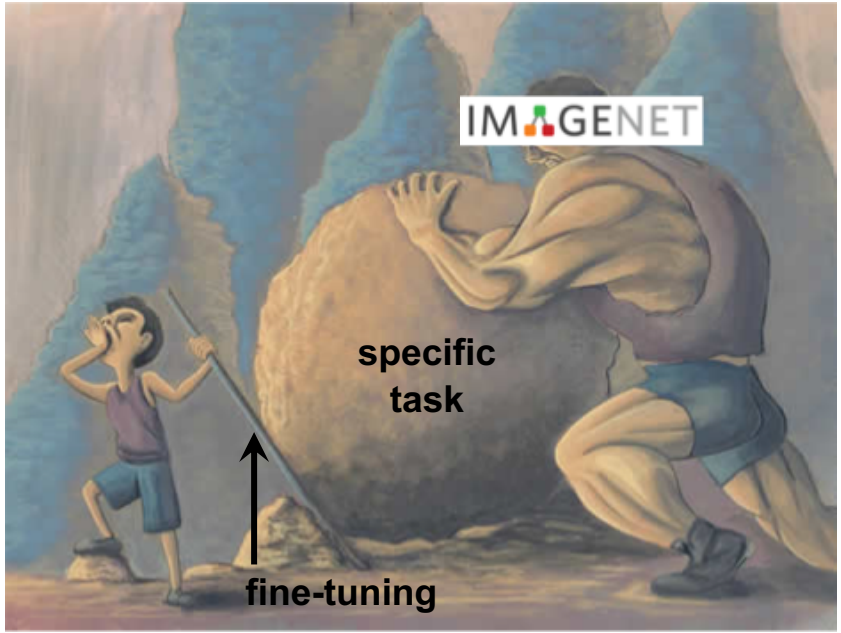
Could we compress the network without losing accuracy on the target domain.

Overview

Domain Transfer and Adaptation

+ Easy transfer using fine-tuning
+ Exploit large pre-trained networks

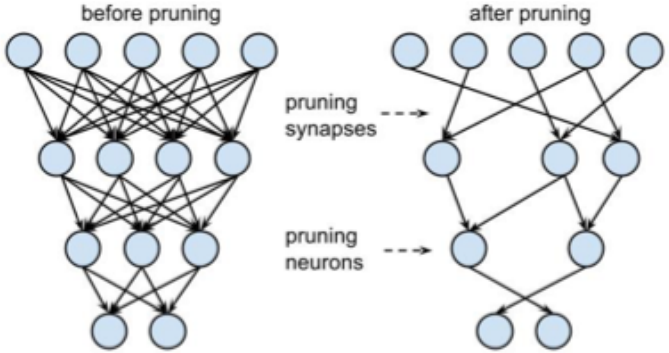
- Network is too large
- Redundant information
- Memory and energy consumption (R.I.P. cell phones, small devices)



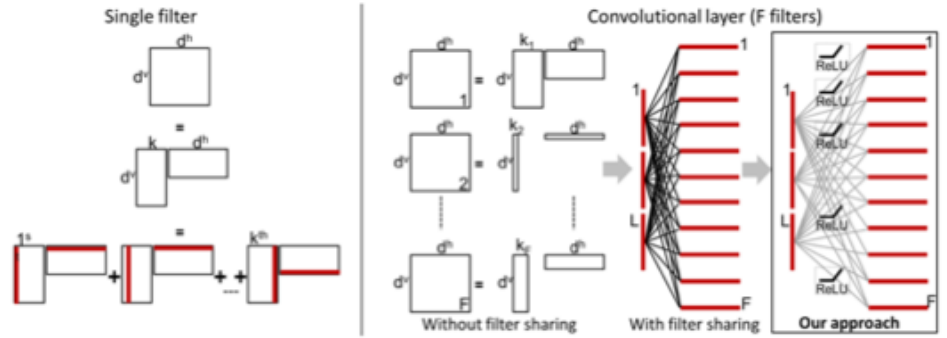
which leads us to… **PRUNING** and **COMPRESSION**

Related Work

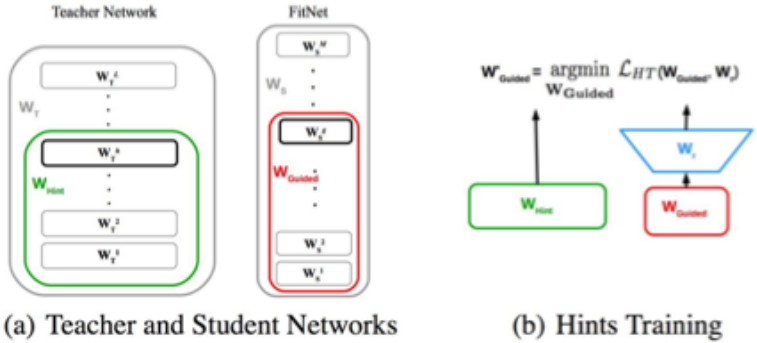
Network pruning



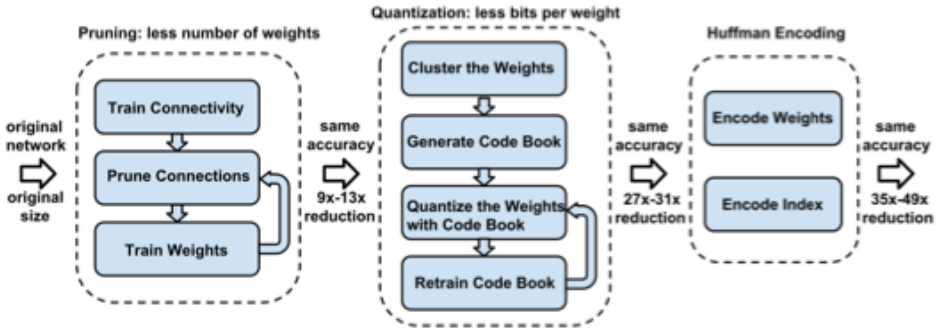
Efficient layer representations



Network Distillation

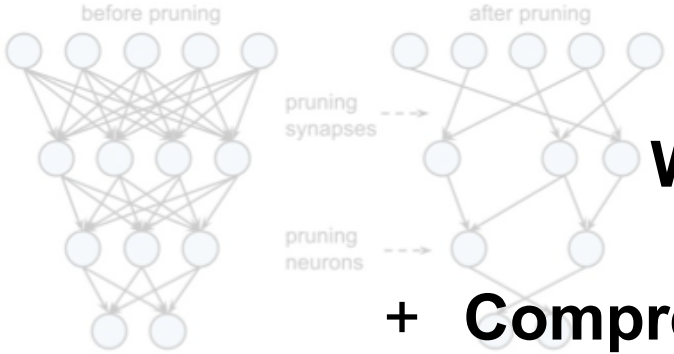


Parameter Quantization

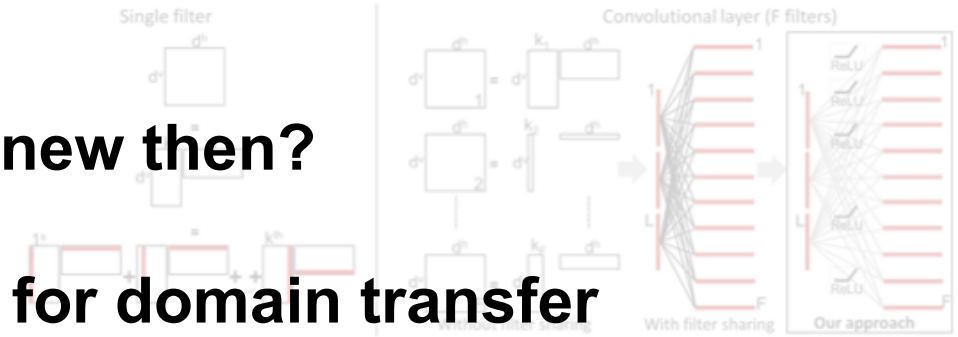


Related Work

Network pruning



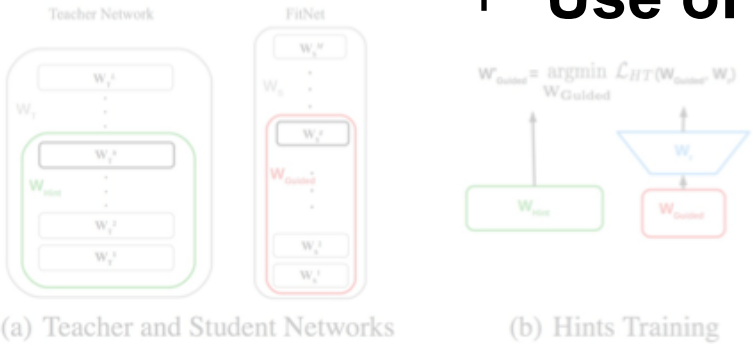
Efficient layer representations



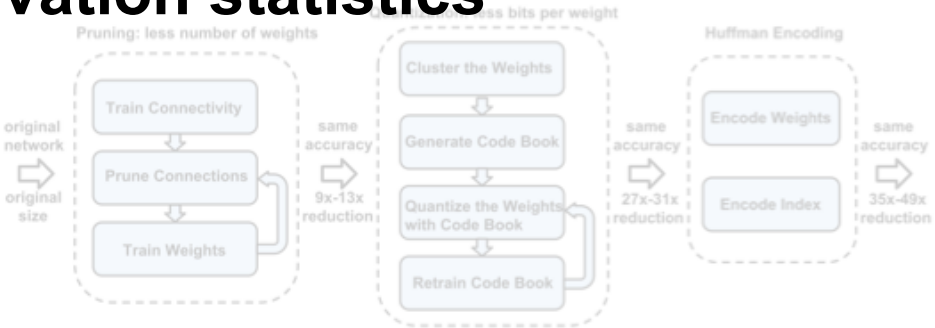
What's new then?

+ Compression for domain transfer

Network Distillation



Parameter Quantization



+ Use of activation statistics

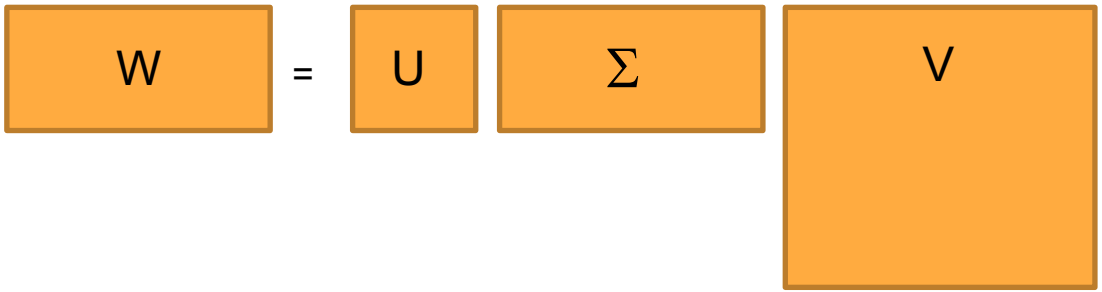
Low Rank Compression

A basic fully connected layer is given by:

$$y = Wx + b,$$

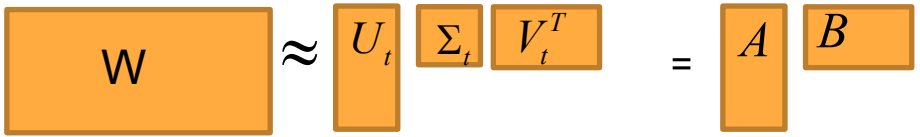
We could compress this layer with SVD, according to:

$$W = U\Sigma V^T$$



where:

$$\approx U_t \Sigma_t V_t^T$$



U: first *t* left singular vectors of W
Σ: diagonal matrix with *t* singular values
V: first *t* right singular vectors of W

J. Xue, J. Li, and Y. Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In Interspeech, 2013.

Low Rank Compression

Truncated SVD basic concept:

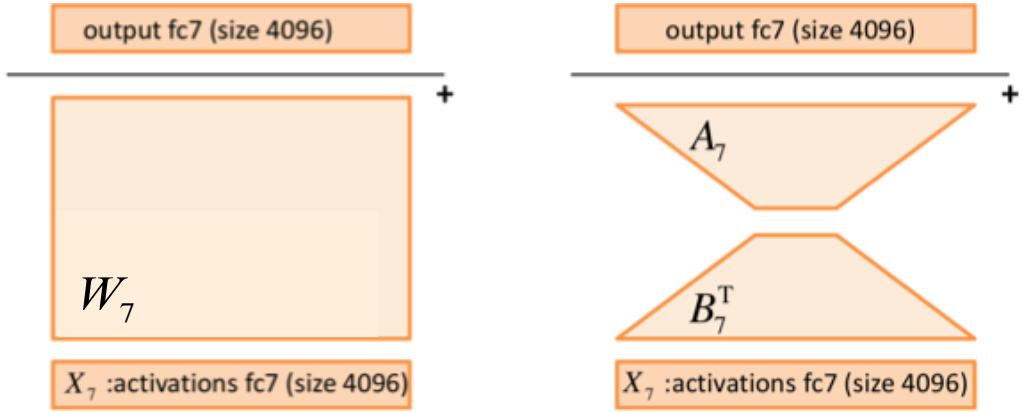
$$y = Wx + b,$$

$$W \approx U\Sigma_t V^T$$

where:

U: first *t* left singular vectors of W
Σ: diagonal matrix with *t* singular values
V: first *t* right singular vectors of W

Our proposal:



Low Rank Compression

Truncated SVD basic concept:

$$y = Wx + b,$$

You only consider the weights and ignore the activations *x*

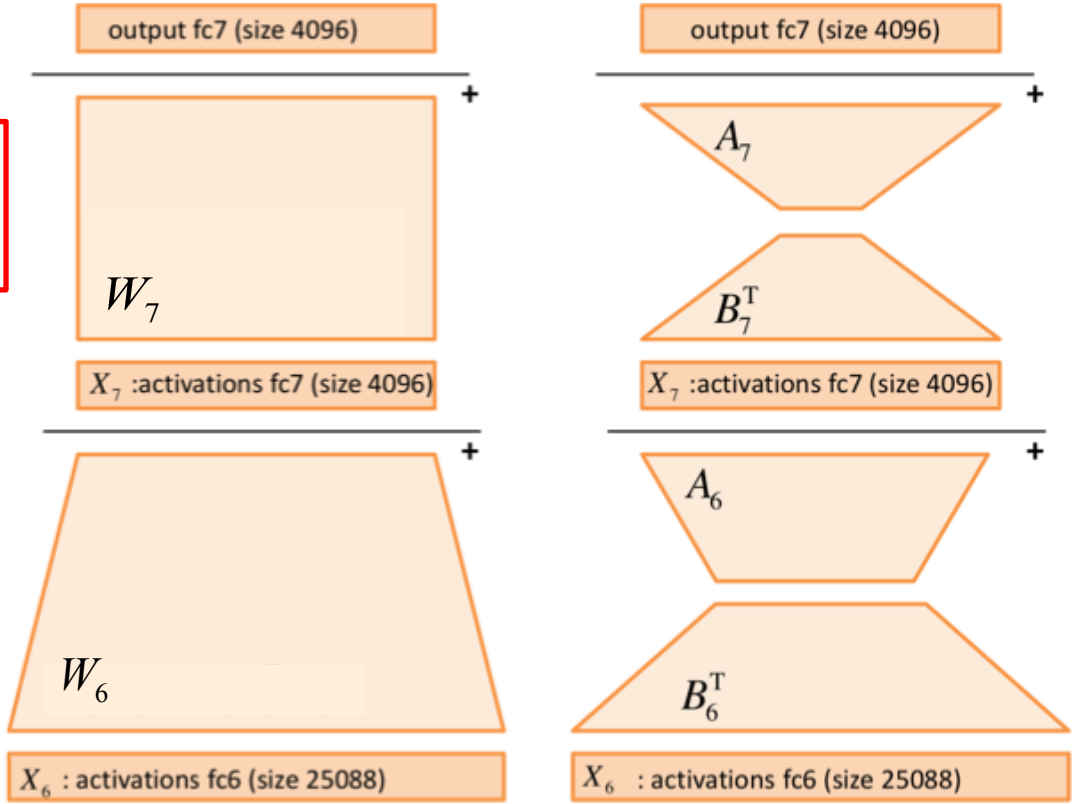
$$W \approx U\Sigma_t V^T$$

where:

U: first *t* left singular vectors of W
Σ: diagonal matrix with *t* singular values
V: first *t* right singular vectors of W

It reduces the #parameter from *uv* to *t(u+v)* and replaces the layer corresponding to W with two new layers.

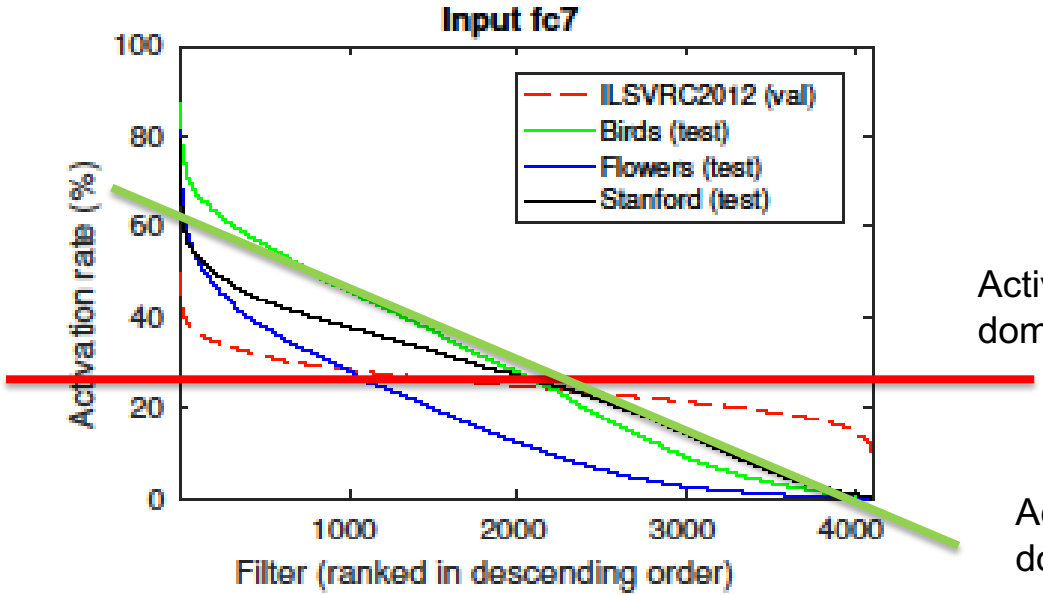
Our proposal:



Motivation

Activation rate → fraction of images in which a neuron has non-zero response.

Is there a shift of the activation distributions in deep networks when changing domain?



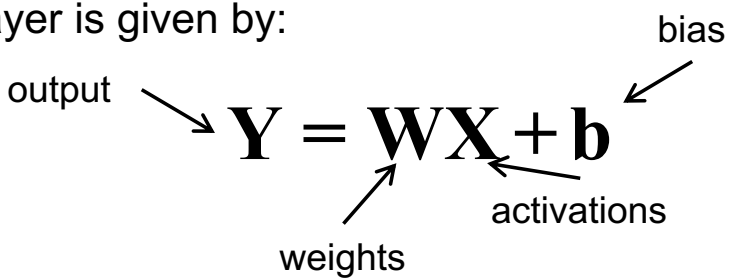
Activations on the source domain are quite uniform.

Activations on the target domain are skewed.

Therefore, it is important to take activation statistics into account.

Domain Adaptive Low Rank (**DALR**) Matrix Decomposition

A Fully-Connected Layer is given by:



Most compression methods focus on compressing **W** : $\left\|\mathbf{W} - \hat{\mathbf{W}}\right\|$

We propose to compress **W** taking activations into account: $\left\|\mathbf{W}\mathbf{X} - \hat{\mathbf{W}}\mathbf{X}\right\| = \left\|\mathbf{W}\mathbf{X} - \mathbf{A}\mathbf{B}^T\mathbf{X}\right\|$

We formalize this as a low-rank regression problem:

$$\arg\min_{C} \left\|\mathbf{Z} - \mathbf{C}\mathbf{X}\right\|_F^2 + \lambda \left\|\mathbf{C}\right\|_F^2$$

$$s.t. rank\left(\mathbf{C}\right) \le k$$

Which has a closed form solution:

$$\mathbf{A} = \mathbf{U}$$

$$\mathbf{B} = \mathbf{U}^{\mathbf{T}}\mathbf{Z}\mathbf{X}^{\mathbf{T}}\left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \boldsymbol{\lambda}\mathbf{I}\right)^{-1}$$

Low Rank Compression

Truncated SVD basic concept:

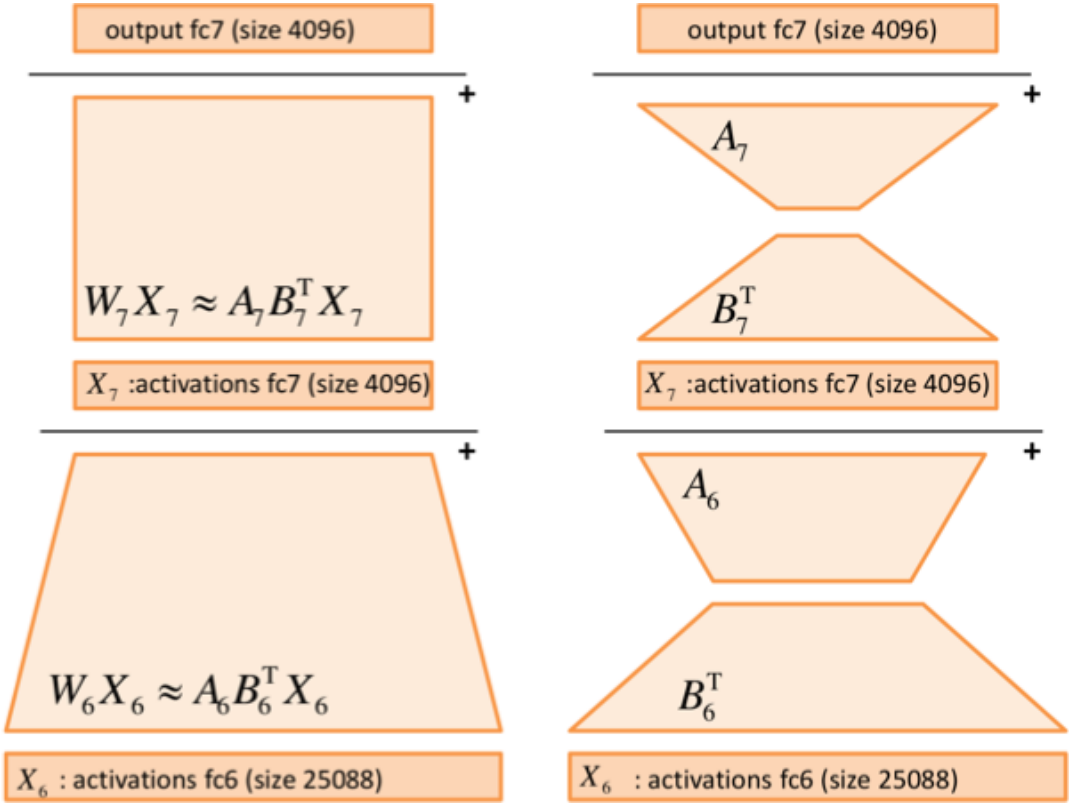
$$y = Wx + b,$$

$$W \approx U\Sigma_t V^T$$

where:

U: first *t* left singular vectors of W
Σ: diagonal matrix with *t* singular values
V: first *t* right singular vectors of W

Our proposal:



Experiments - Compression on Source Domain

dim kept	32	64	128	256	512	1024
params	0.91%	1.82%	3.64%	7.27%	14.54%	29.08%
SVD	79.44	50.82	36.44	34.80	34.40	**34.18**
SVD + BC	73.41	46.54	36.21	34.82	34.33	34.22
DALR	**66.43**	**44.50**	**36.06**	**34.63**	**34.28**	34.20

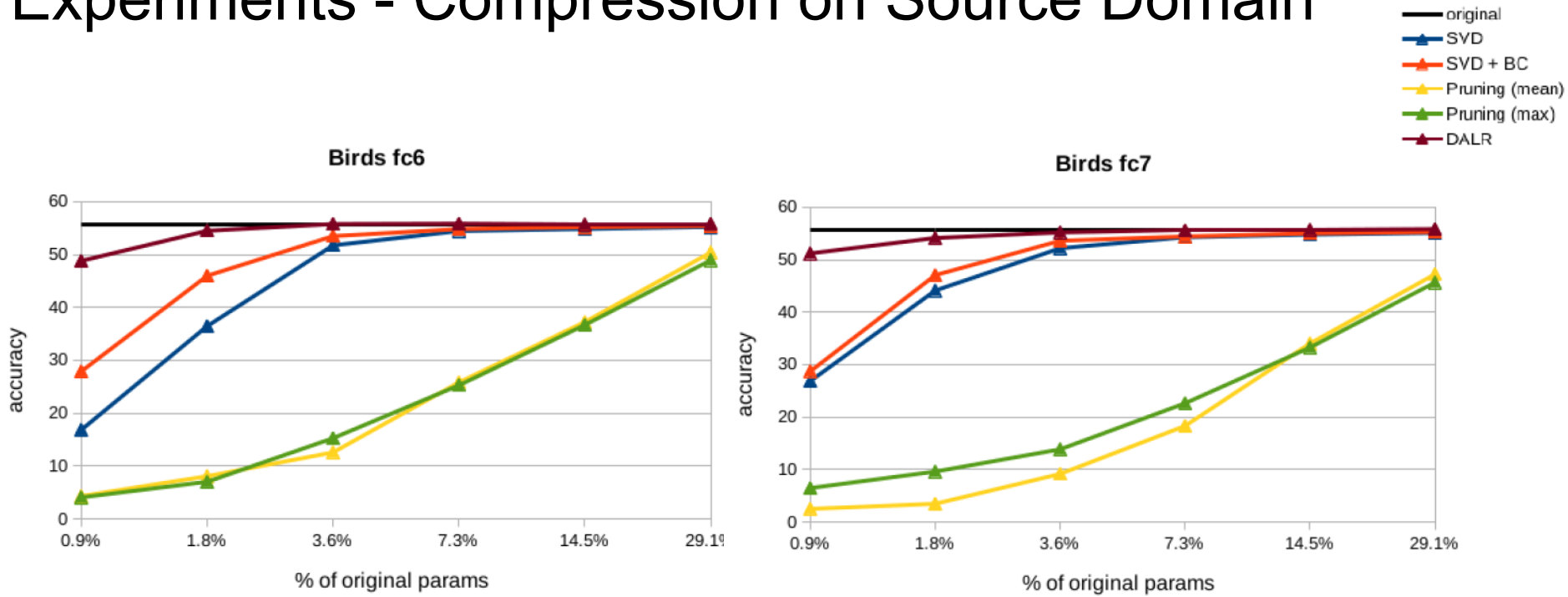
Table 1. Top-1 error rate results on ImageNet for fc6. We report the dimensions kept k and the percentage of parameters compressed. The uncompressed top-1 error rate is 34.24%.

dim kept	32	64	128	256	512	1024
params	1.56%	3.13%	6.25%	12.5%	25%	50%
SVD	57.07	40.68	35.50	34.63	34.40	34.35
SVD + BC	**55.57**	**39.75**	**35.14**	**34.51**	**34.35**	**34.30**
DALR	56.32	40.25	35.26	34.54	34.40	34.33

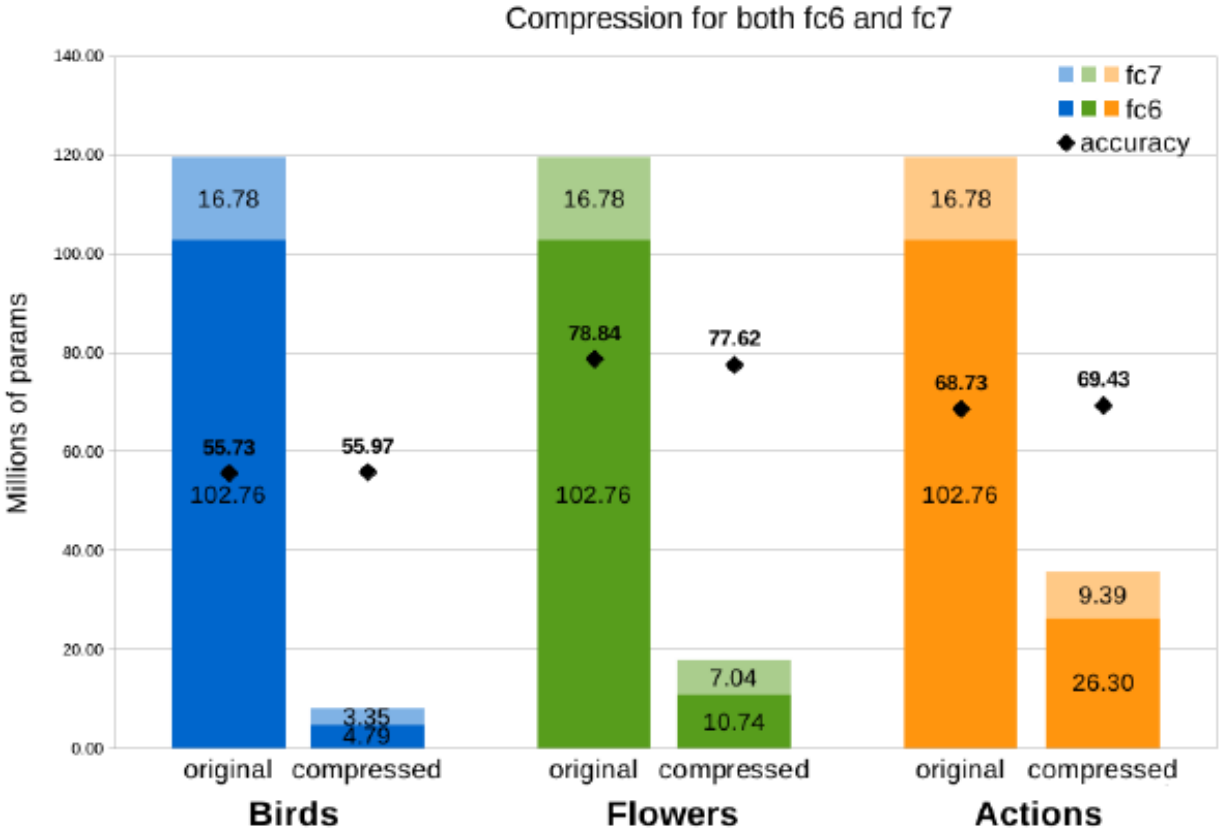
Table 2. ImageNet fc7 - uncompressed top-1 error rate: 34.24%.

Proposed methods outperform SVD, but by a small margin.

Experiments - Compression on Source Domain



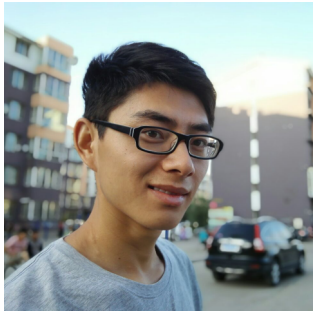
Experiments – of both FC layers



Learning from Rankings

Xialei Liu, Joost van de Weijer, Andrew Bagdanov

ICCV 2017
CVPR 2018 (under submission)

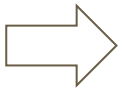


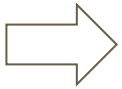
LAMP The Learning and Machine Perception Team

CVC Centre de Visió per Computador

Image quality assessment (IQA)

The goal of IQA is to design algorithms for *objective* evaluation of quality in a way that is consistent with *subjective* human evaluation.





score: 85 ∈ [0,100]

Contributions

Ranked sets

Siamese Network

Learn from rankings

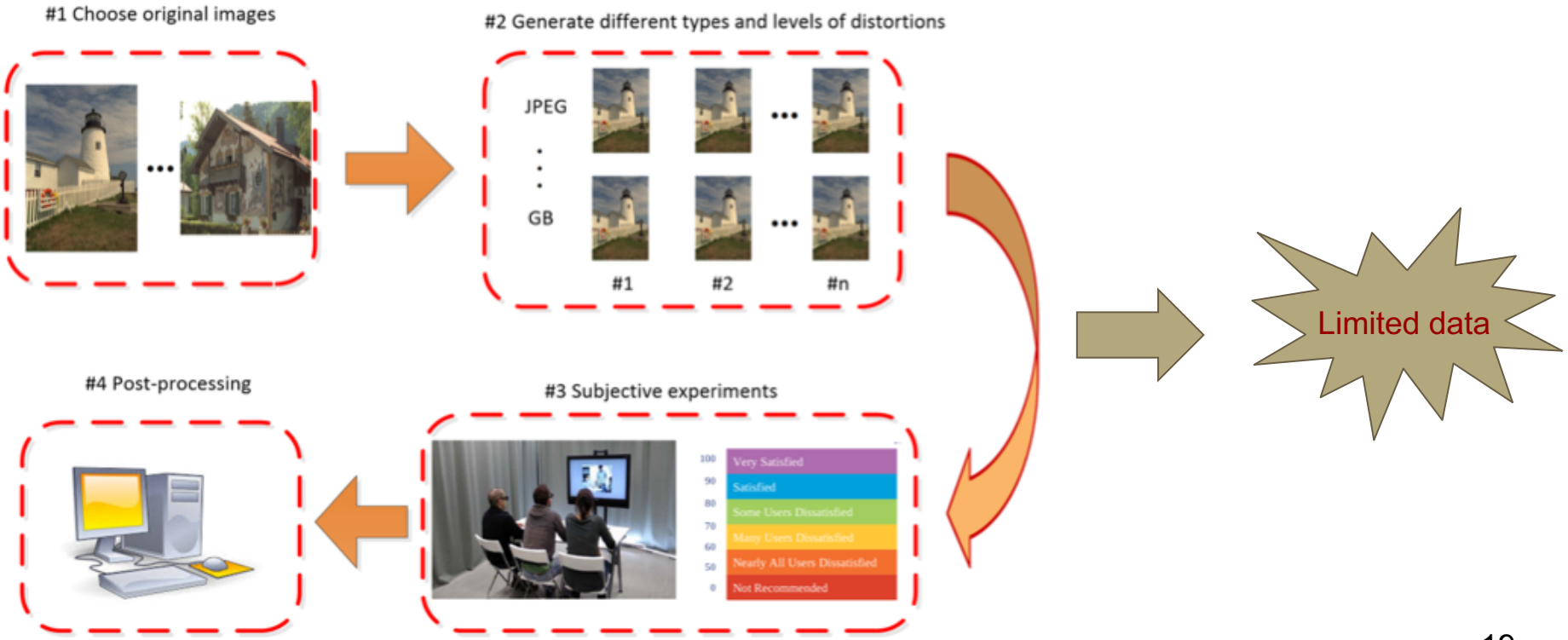
Speed-up

Fast Siamese backpropagation

Image quality assessment

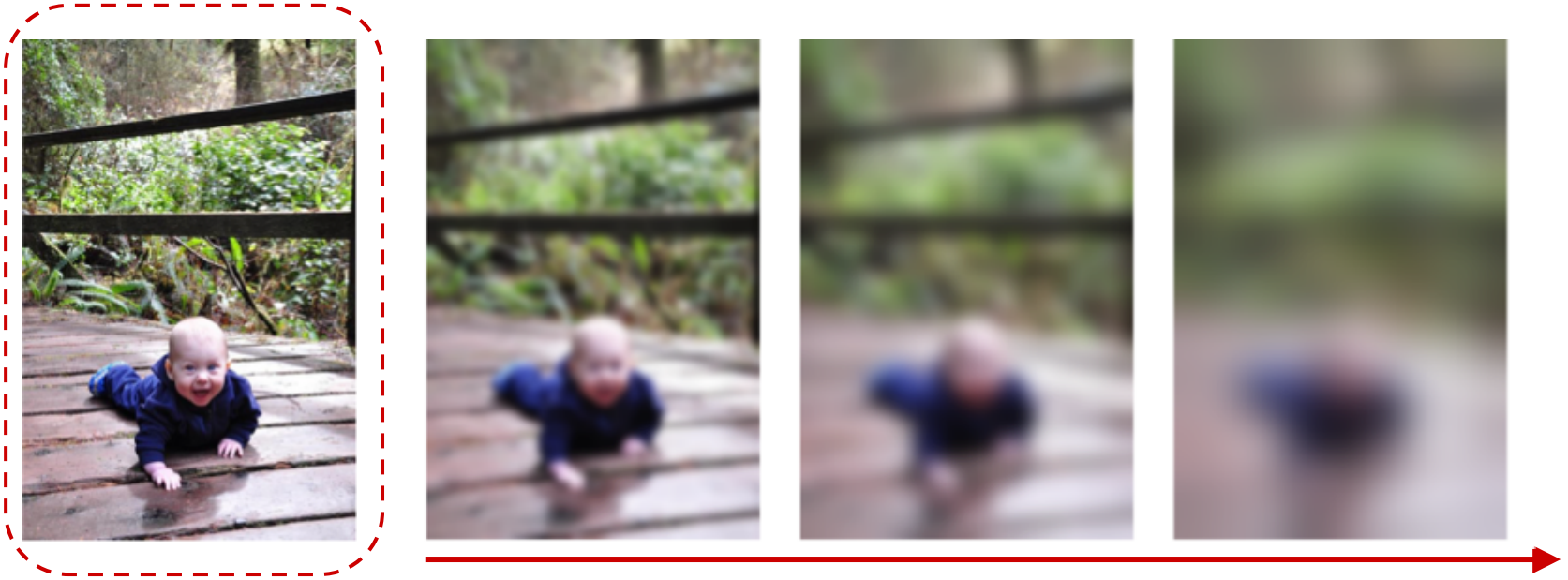
Start-of-the-art performance

How existing IQA datasets are generated?



Our approach--learn from rankings

- Large human annotated IQA data is difficult to obtain
- Easy to generate images that are ranked according to their image quality.



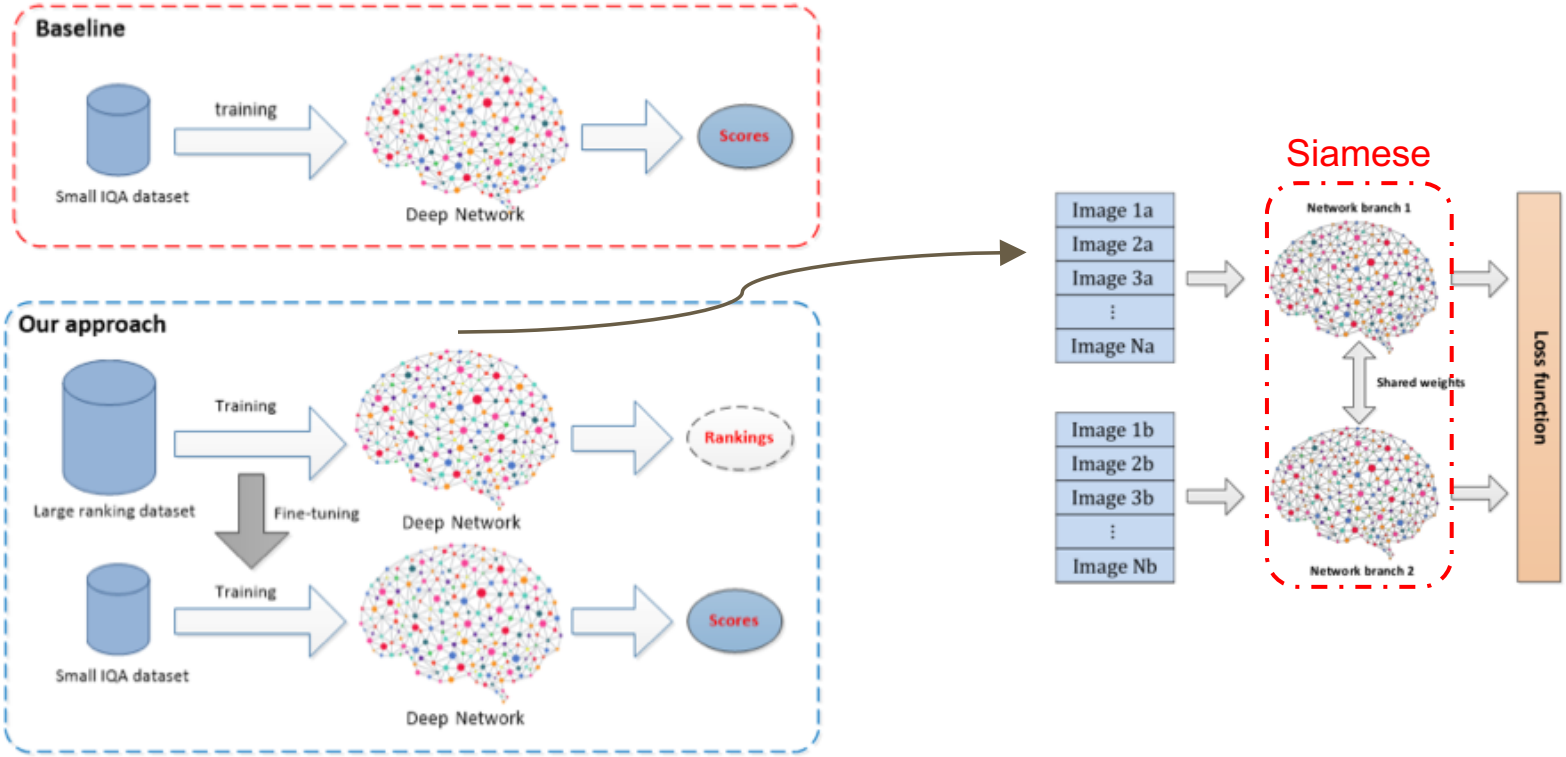
Original image

Distortion level 1

Distortion level 2

Distortion level 3

Our approach--learn from rankings



Siamese Network--loss function

- **Contrastive loss function**

Similarity Margin Distance

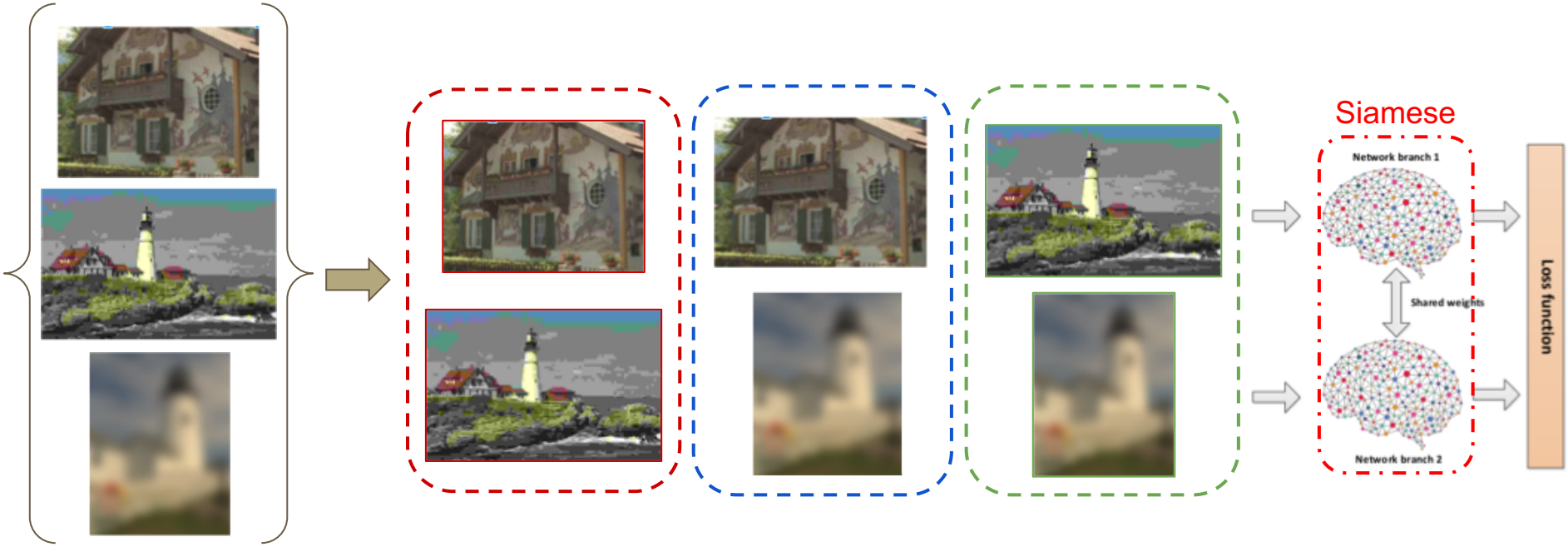
$$L\left(x_1, x_2, l\right) = \frac{1}{2} l D(x_1, x_2)^2 + \frac{1}{2}\left(1 - l\right) \max\left(0, \varepsilon - D(x_1, x_2)\right)^2$$

- **Ranking loss function**

Distance

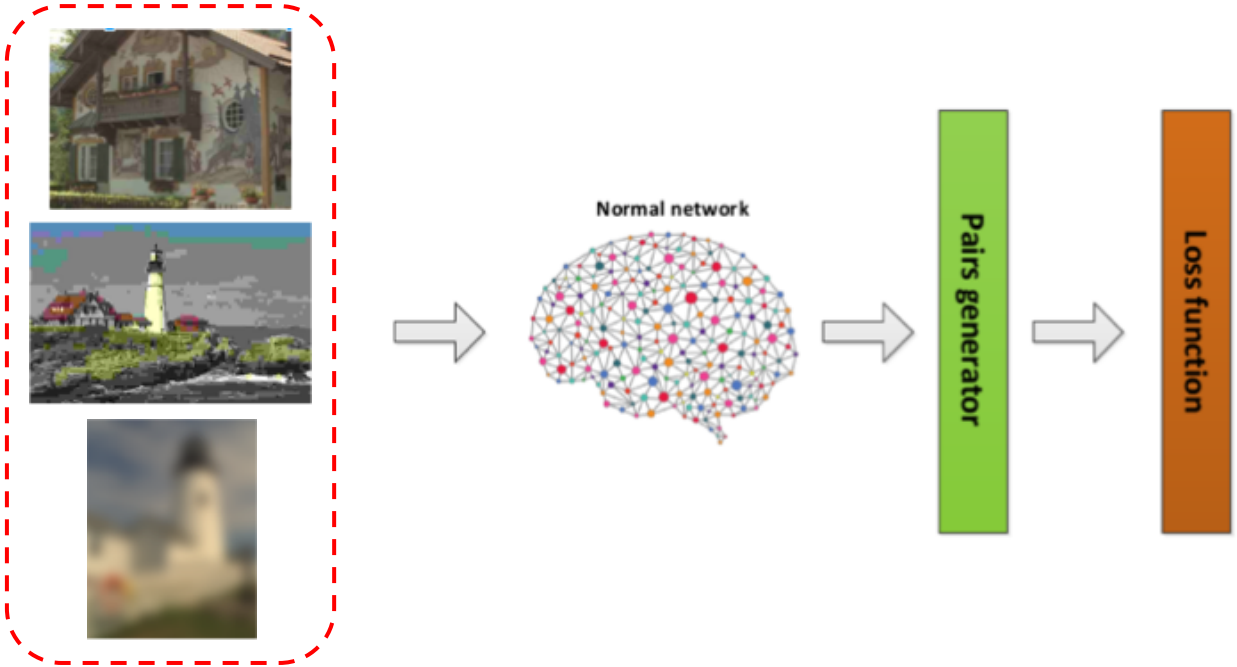
$$L(x_1, x_2) = \max\left(0, -\left(f(x_1) - f(x_2)\right) + \varepsilon\right)$$

Standard Siamese Network--redundancy



Conclusion: To compute the backpropagation of 3 pairs, 3 images are passed through the network twice.

Our proposal: Fast Siamese Network



Conclusion: To compute the backpropagation of 3 pairs, 3 images are passed through the network once.

How fast Siamese Network works?

- ***n*** levels of distortions
- passes are $n^2 - n$

- ***n*** levels of distortions
- passes are ***n***

- ***n-1*** times
- ***n=M*** (mini-batch size)

Experimental setup

Generation of ranked images: Waterloo and Places2 dataset

- JPEG Compression
- JPEG2000 Compression
- Gaussian Blur
- Gaussian Noise

Each original image is distorted by 5 levels for 4 distortion type (1 image -> 20 distorted image)

Network architecture

Blocks	Output sizes	Details of block
Input block	3x227x227	3-channels images
Block 1	32x57x57	conv1(3x3), relu1, pool1(4x4)
Block 2	32x14x14	conv2(3x3), relu2, pool2(4x4)
Block 3	32x12x12	conv3(3x3), relu3
Block 4	32x1x1	conv4(12x12)
Block 5	1	fc1

⇔ Shallow AlexNet VGG-16

Experimental setup

- **Training:** randomly crop from the original images
- **Testing:** average pooling is used to obtain the final results

Evaluation protocols

- **Linear Correlation Coefficient (LCC):** linear correlation between the ground truth and the predicted quality scores

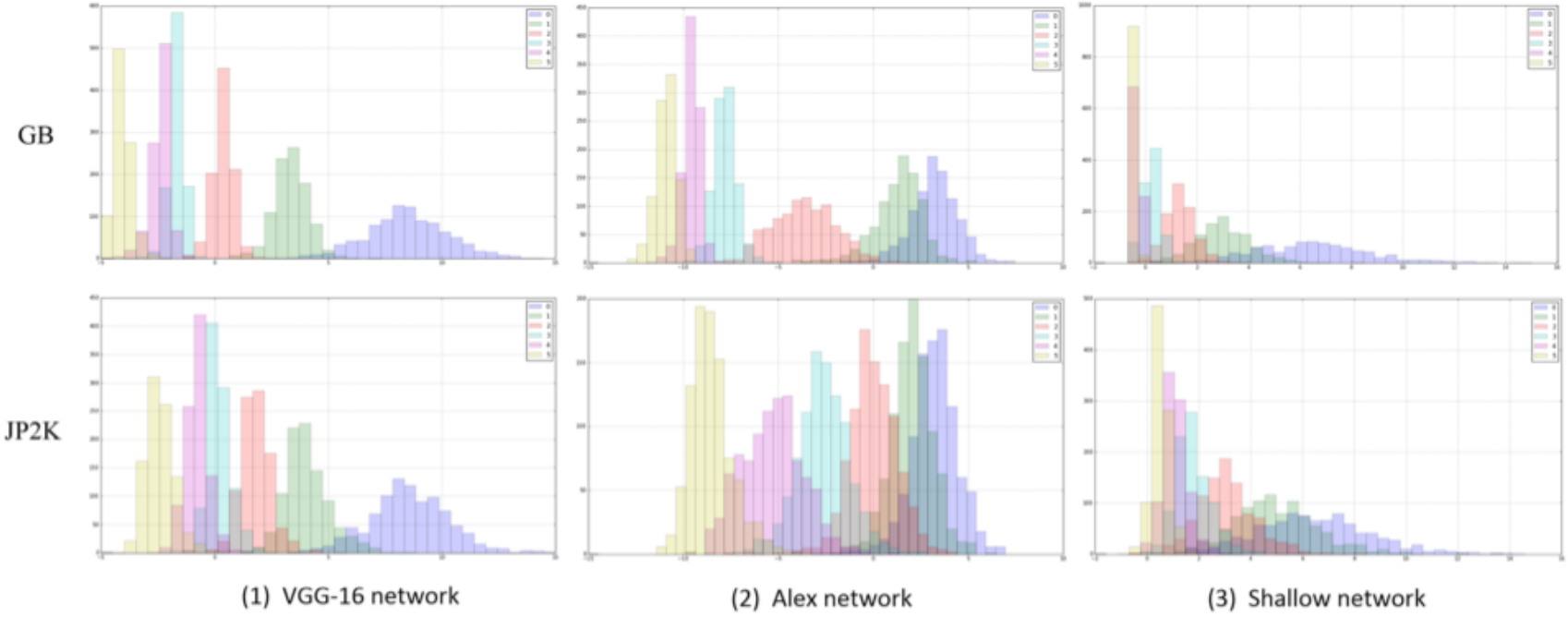
$$LCC = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i}^{N} (x_i - \overline{x})^2} \sqrt{\sum_{i}^{N} (y_i - \overline{y})^2}}$$

- **Spearman Rank Order Correlation Coefficient (SROCC):** monotonic relationship between the ground truth and the predicted quality scores

$$SROCC = 1 - \frac{6 \sum_{i=1}^{N} (v_i - p_i)^2}{N(N^2 - 1)}$$

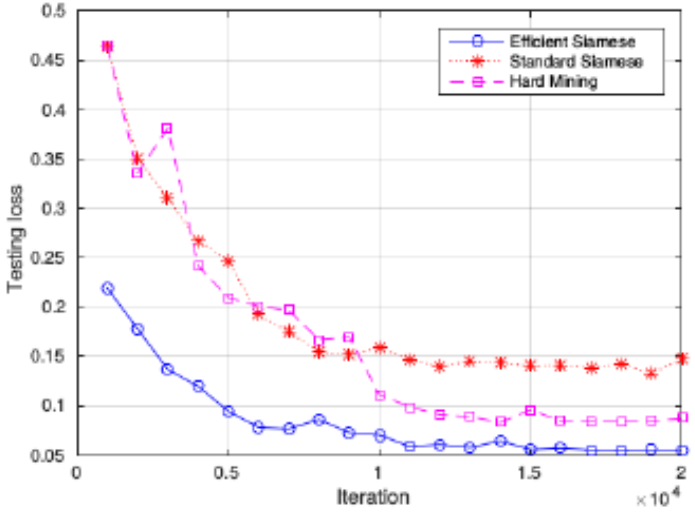
1.General settings

- **Varying the network architecture**

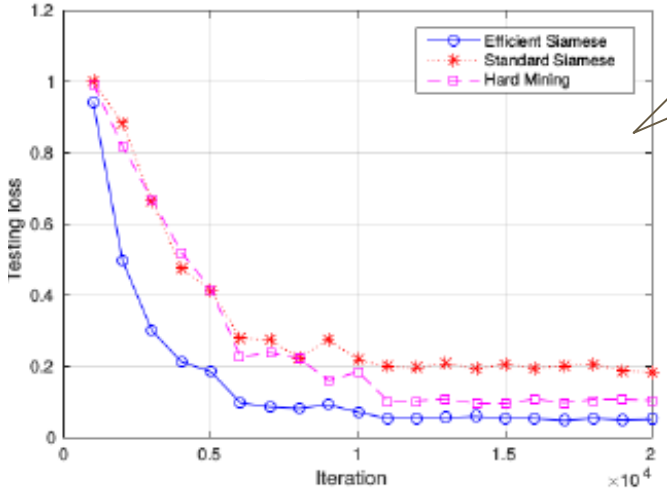


Conclusion: VGG-16 performs better for ranking different levels of distortions

2. Standard Siamese v.s. fast Siamese network



(a) Ranking loss on JP2K



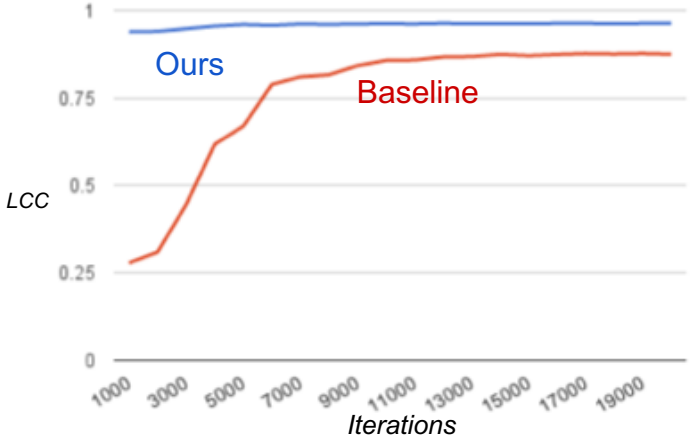
(b) Ranking loss on JPEG

- AlexNet
- Same mini-batch
- JPEG distortion
- 10,000 iteration

3. Comparison of our approach with baseline

LCC evaluation

	Shallow	AlexNet	VGG-16
Baseline	0.831	0.801	0.915
Ours	0.930	0.941	0.970



- The baseline corresponds to results by models trained from scratch
- Our approach means the results from models fine-tuned from ranking network

Conclusion: Our approach boosts the performance of IQA problem

4. Comparison of our approach with state-of-the-art

Evaluation on LIVE: randomly split LIVE dataset into 80% training samples and 20% testing samples and repeat the process 10 times. Average of LCC and SROCC are the final results.

LCC	JP2K	JPEG	GN	GB	FF	ALL
PSNR	0.873	0.876	0.926	0.779	0.87	0.856
SSIM	0.921	0.955	0.982	0.893	0.939	0.906
PSIM	0.91	0.985	0.976	0.978	0.912	0.96
DIVINE	0.922	0.921	0.988	0.923	0.888	0.917
BLIIDNS-II	0.935	0.968	0.98	0.938	0.896	0.93
BRISQUE	0.923	0.973	0.985	0.951	0.903	0.942
CORNIA	0.951	0.965	0.987	0.968	0.917	0.935
CNN	0.953	0.981	0.984	0.953	0.933	0.953
SOM	0.952	0.961	0.991	0.974	**0.954**	0.962
Ours	**0.976**	**0.987**	**0.995**	**0.988**	–	**0.970**

*Zhang P, Zhou W, Wu L, et al. **SOM**: Semantic obviousness metric for image quality assessment[C]//CVPR. 2015: 2394-2402.

Conclusions

1) Taking the activations of the target domain into account allows for significant higher compression rates

2) Results show that the method can compress to an additional factor of 2-4 over SVD.

3) Rankings can be used ot increase the datasets, e.g. in quality assesment

4) Apply fast Siamese backpropagation to solve pairs selection redundancy leads to better loss optimization

The end!

Thank you very much!