ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Deep Learning

Margonis Phevos

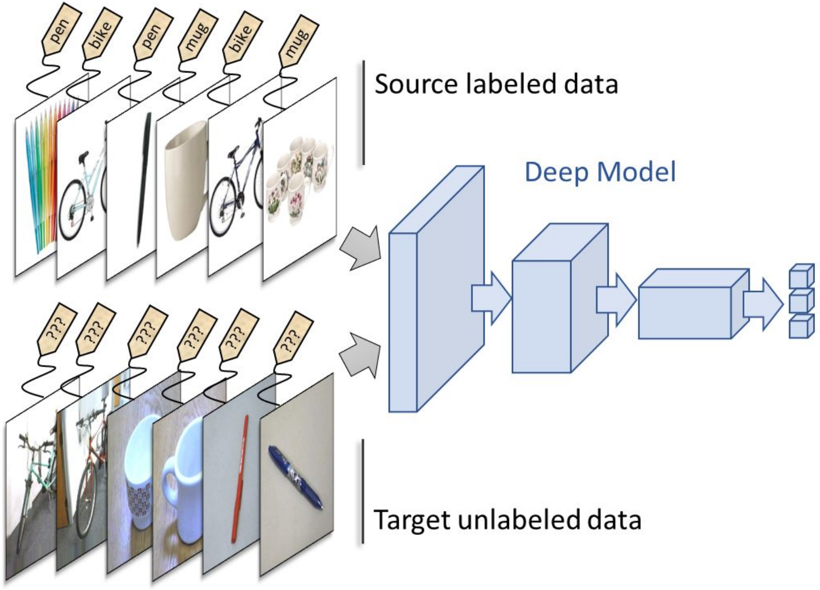
June 2024

Contents

Introduction

Problem Definition

- What is Domain Adaptation?
- Large labeled *source* dataset
- Small unlabeled *target* dataset
- Learn domain invariant features
- Generalize better to the target domain
- Perform two experiments with one algorithm

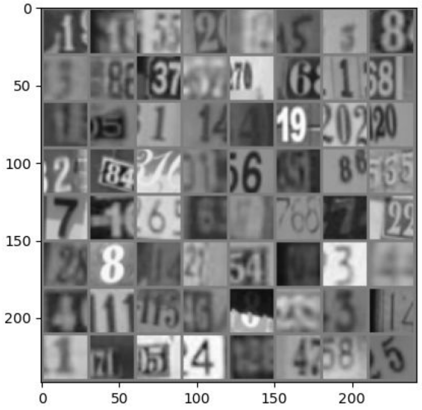


Data sets

Experiment 1 - SVHN to MNIST

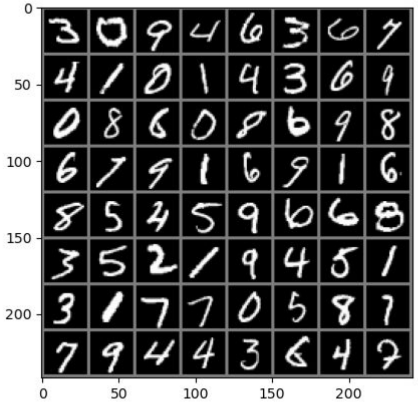
SVHN:

- 100.000 RGB images
- 32 x 32
- Transformations



MNIST:

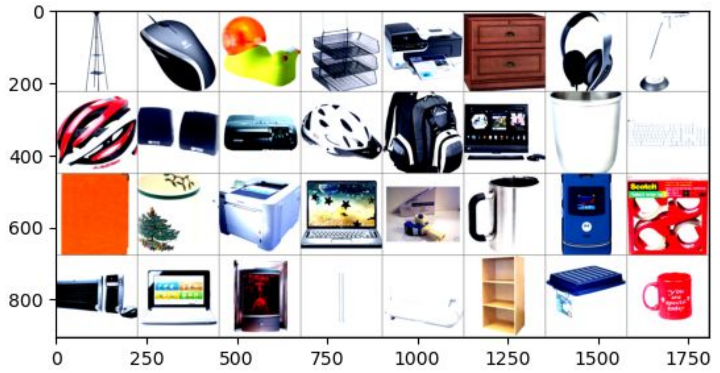
- 70.000 Gray scale images
- 28 x 28
- Normalize



Experiment 2 - Amazon to Webcam

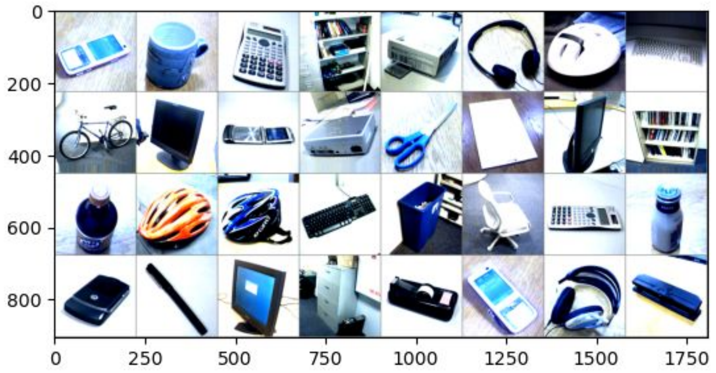
Amazon:

- 2.817 RGB images
- 300 x 300
- Transformations (Resize, C.Crop, Norm)



Webcam:

- 795 RGB images
- 640 x 480
- Transformations (Resize, C.Crop, Norm)



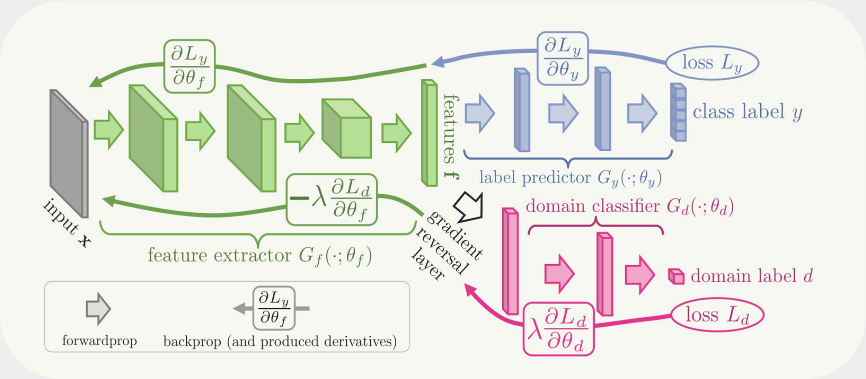
Baselines

1. No Adaptation
2. Paper Implementation
3. Train on Target

Domain-Adversarial Neural Network (DANN)

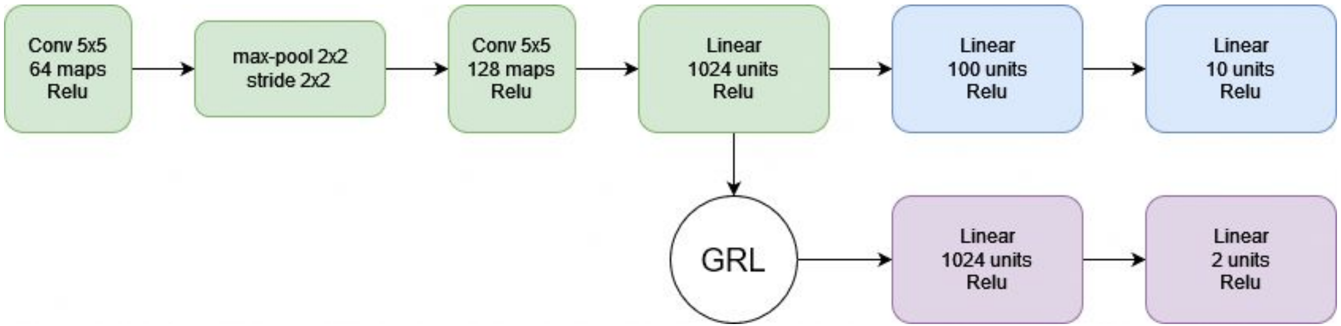
Model

1. Feature Extraction
2. Label Prediction
3. Domain Classification
4. Adversarial Training



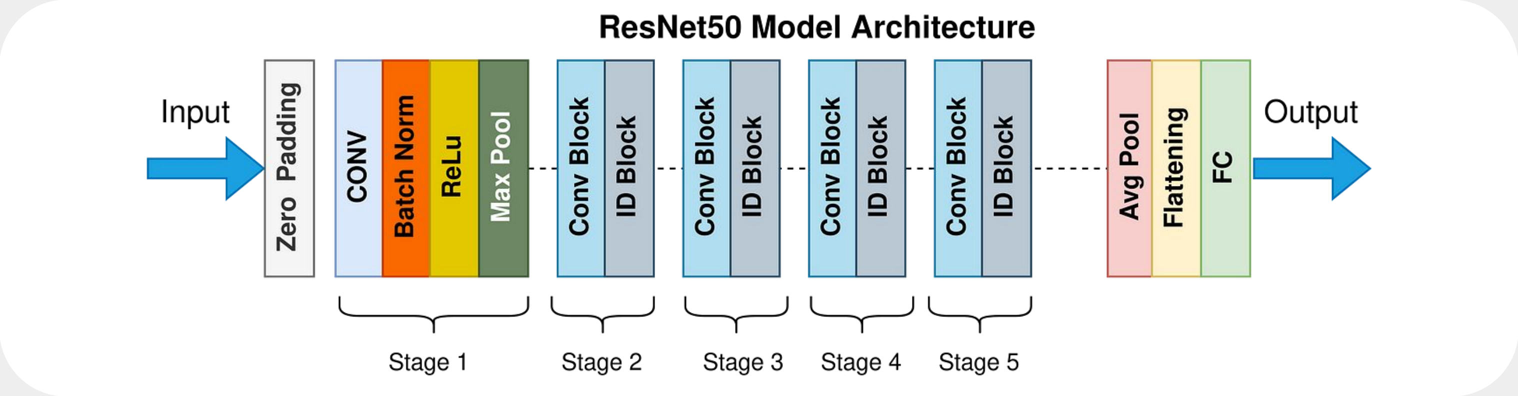
Feature extractor for the *Digits* experiment

A modified LeNet, with dropout and Batch Norm.



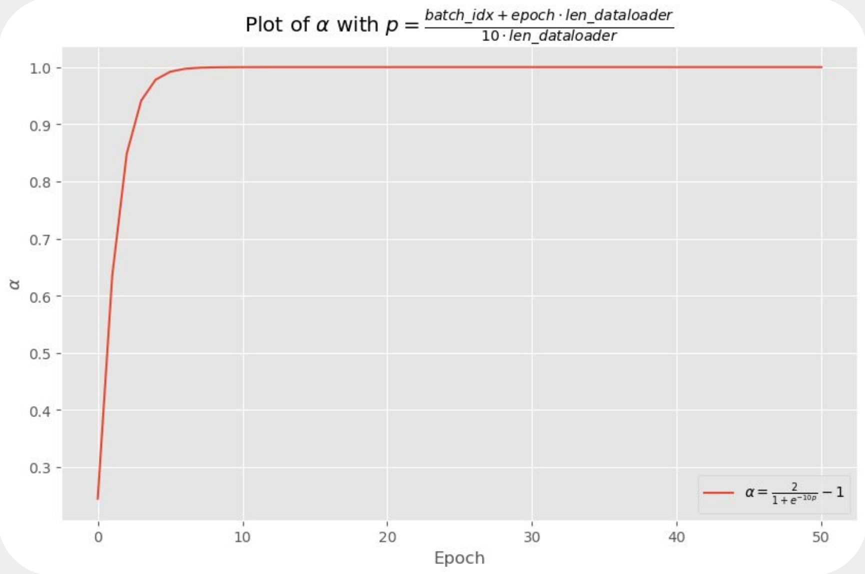
Feature extractor for the *Office-31* experiment

A pretrained ResNet 50



GRL

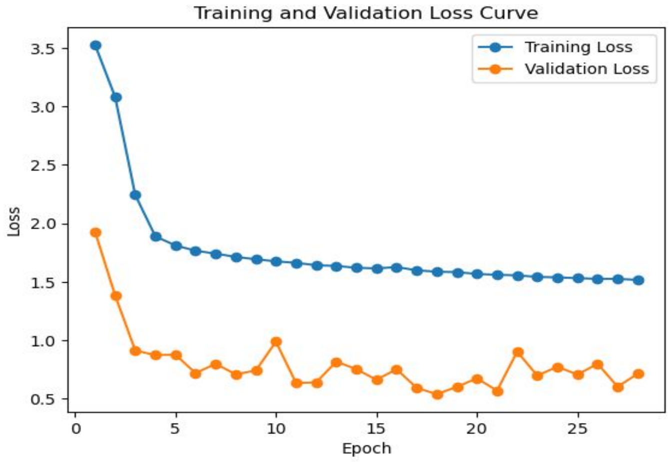
The **alpha (λ)** parameter in GRL adjusts gradient reversal, balancing domain learning and accuracy.



Criterion and Optimizers

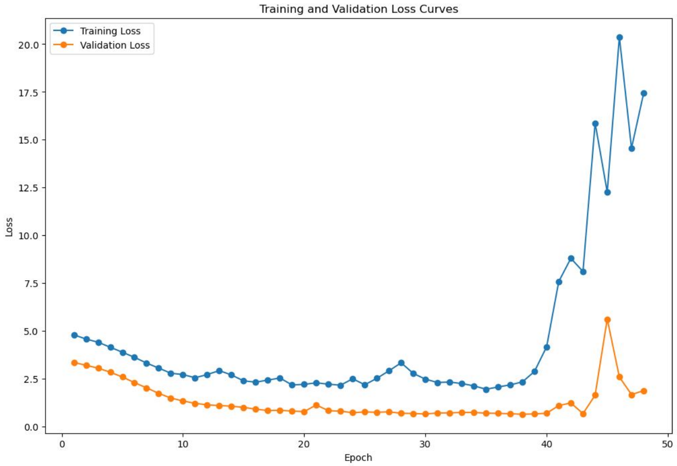
SVHN to MNIST:

- Cross Entropy Loss
- Adam optimizer
  - Learning rate = 0.001



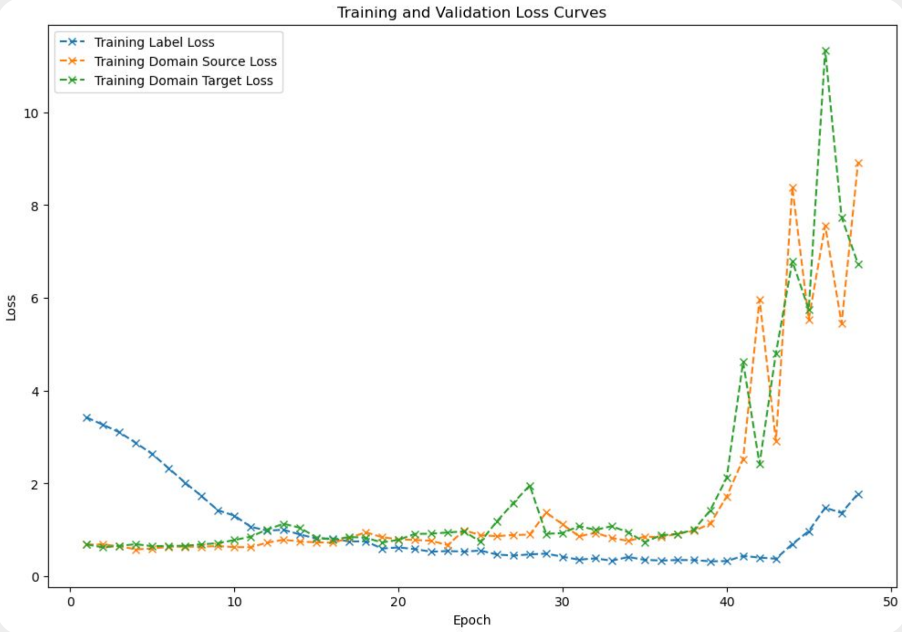
Amazon to Webcam:

- Cross Entropy Loss
- SGD optimizer
  - Learning rate = 0.001
  - Momentum = 0.9



Overfit?

Adversarial training breakdown



Results

Results

We outperform the two baselines

Method	Accuracy	
	SVHN MNIST	**Amazon Webcam**
No Adaptation	70%	74%
Paper Implementation	74%	73%
Our DANN	**78%**	**80%**
Train on Target	99%	99%

Failed Experiment

Correlation Alignment (CORAL)

Algorithm 1: Correlation Alignment (CORAL)

Output: Aligned source feature matrix $\mathbf{X}_s'$

Compute the covariance matrices:
$\mathbf{C_s} = \mathrm{cov}(\mathbf{X_s}),\ \mathbf{C_t} = \mathrm{cov}(\mathbf{X_t});$

Regularize the covariance matrices:
$\mathbf{C}_s' = \mathbf{C_s} + \lambda\mathbf{I},\ \mathbf{C}_t' = \mathbf{C_t} + \lambda\mathbf{I};$

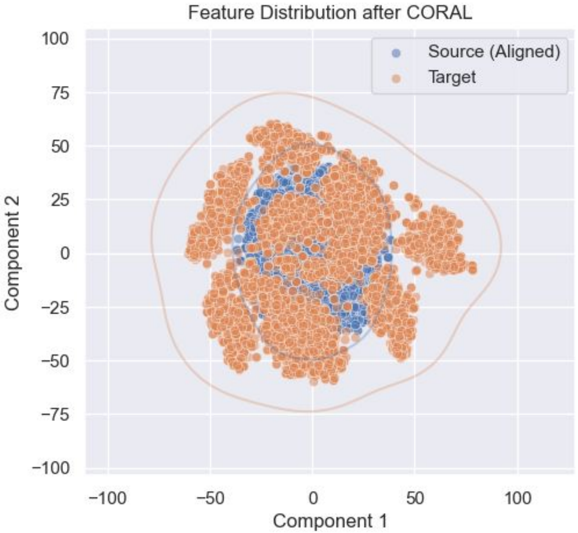
Compute the square root inverse of $\mathbf{C}_s'$:
$\mathbf{C}_s^{-\frac{1}{2}} = \mathbf{C}_s'^{-\frac{1}{2}};$

Compute the square root of $\mathbf{C}_t'$: $\mathbf{C}_t^{\frac{1}{2}} = \mathbf{C}_t'^{\frac{1}{2}};$

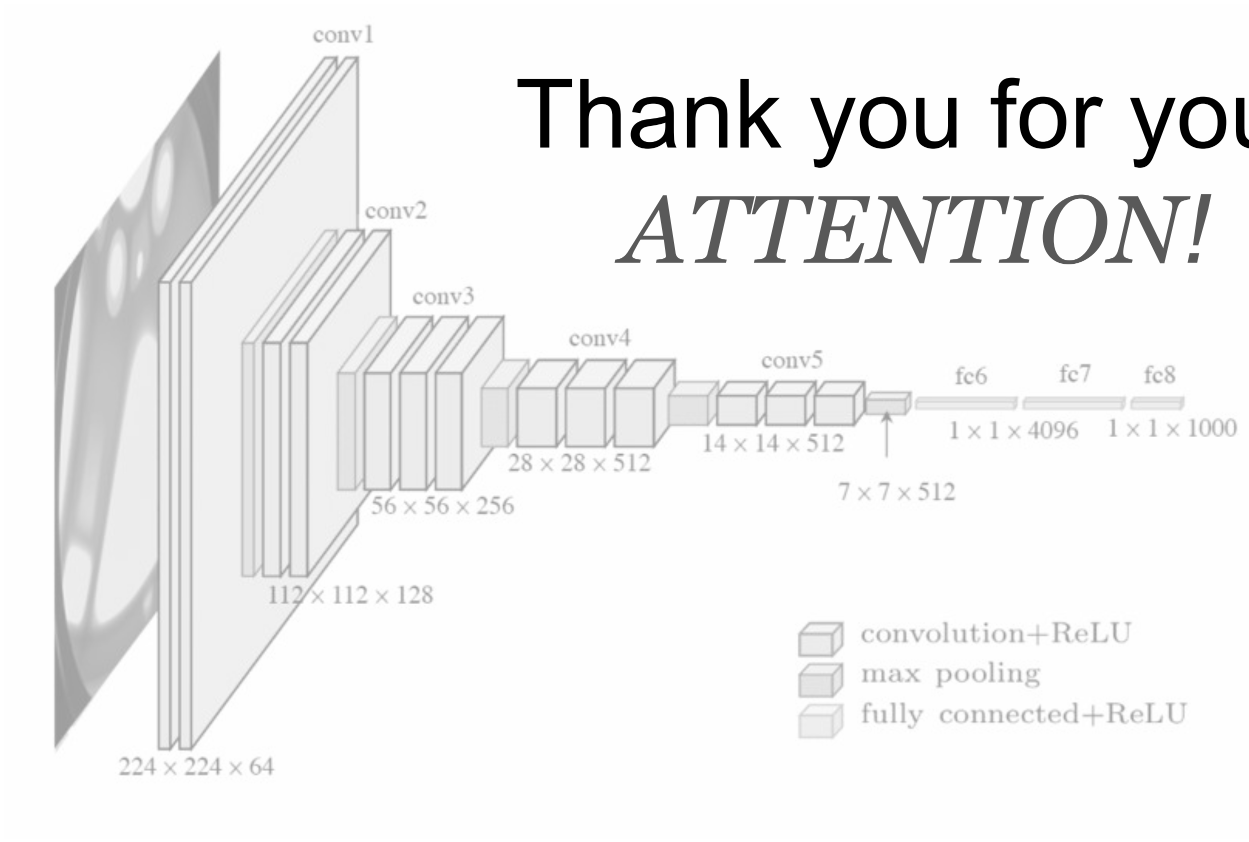
Remove complex components if any:
$\mathbf{C}_s^{-\frac{1}{2}} = \Re(\mathbf{C}_s^{-\frac{1}{2}}),\ \mathbf{C}_t^{\frac{1}{2}} = \Re(\mathbf{C}_t^{\frac{1}{2}});$

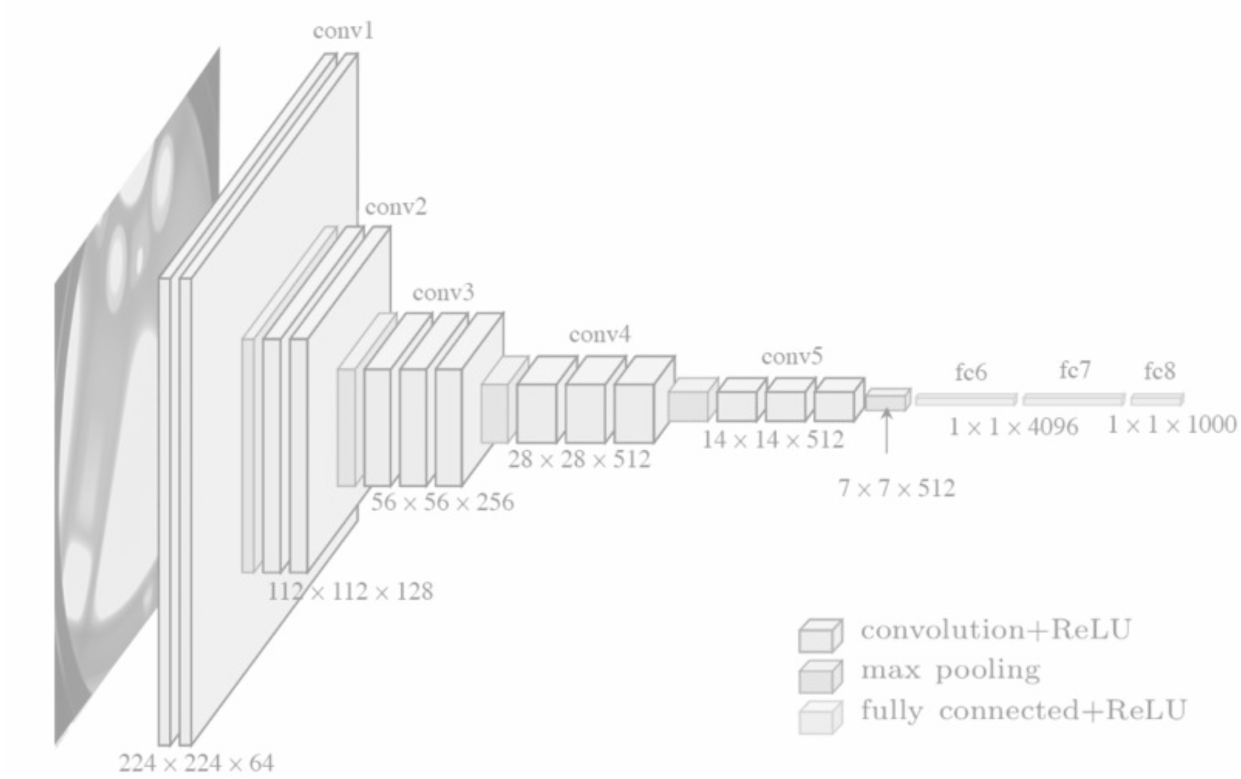
Align the source features: $\mathbf{X}_s' = \mathbf{X}_s\mathbf{C}_s^{-\frac{1}{2}}\mathbf{C}_t^{\frac{1}{2}};$

return $\mathbf{X}_s'$



Thank you for your *ATTENTION!*




conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

Discussion…