# Beyond Sharing Weights for Deep Domain Adaptation

Mathieu Salzmann
EPFL – CVLab

Joint work with A. Rozantsev and P. Fua

# Domain Adaptation for Visual Recognition

Training data (source domain)

Test data (target domain)

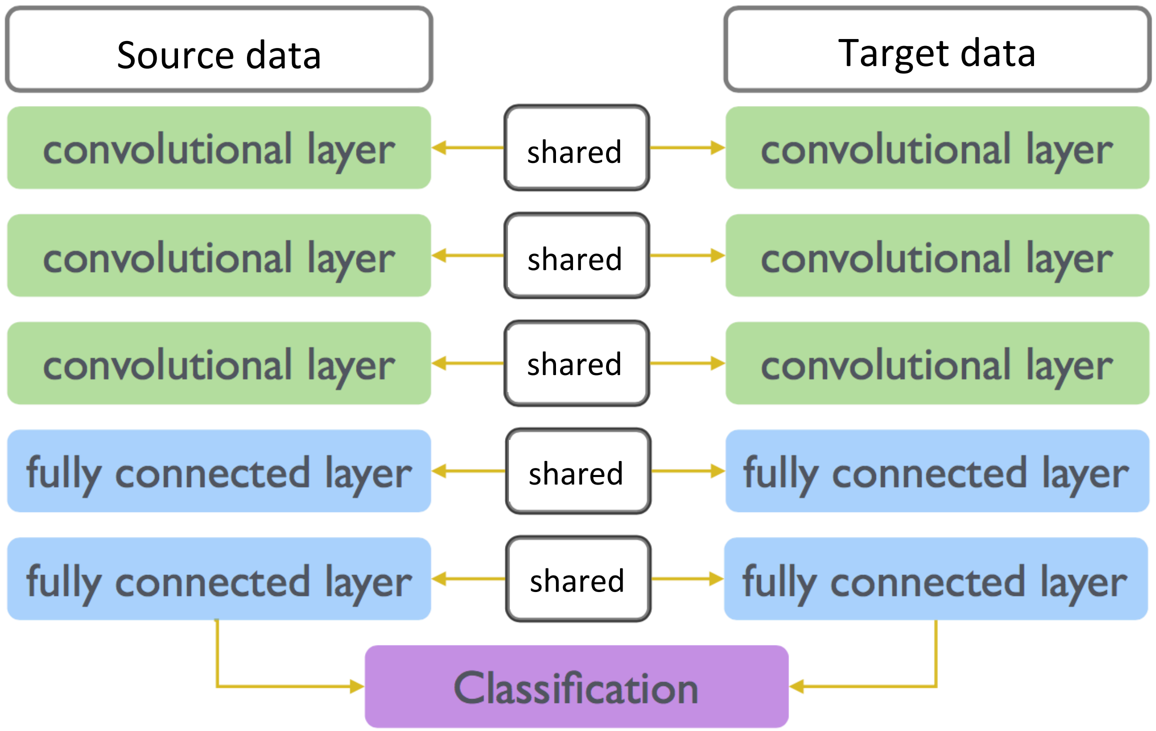
A classifier trained on one domain may perform poorly on another domain

# Deep Domain Adaptation: Existing Approaches

- Siamese network
  - Chopra et al., CVPR 2005
- MMD loss
  - Tzeng et al., arXiv 2014
  - Long et al., ICML 2015
- Domain classifier
  - Ganin & Lempitsky, ICML 2015
  - Tzeng et al., ICCV 2015

# Existing Approaches: Sharing Weights

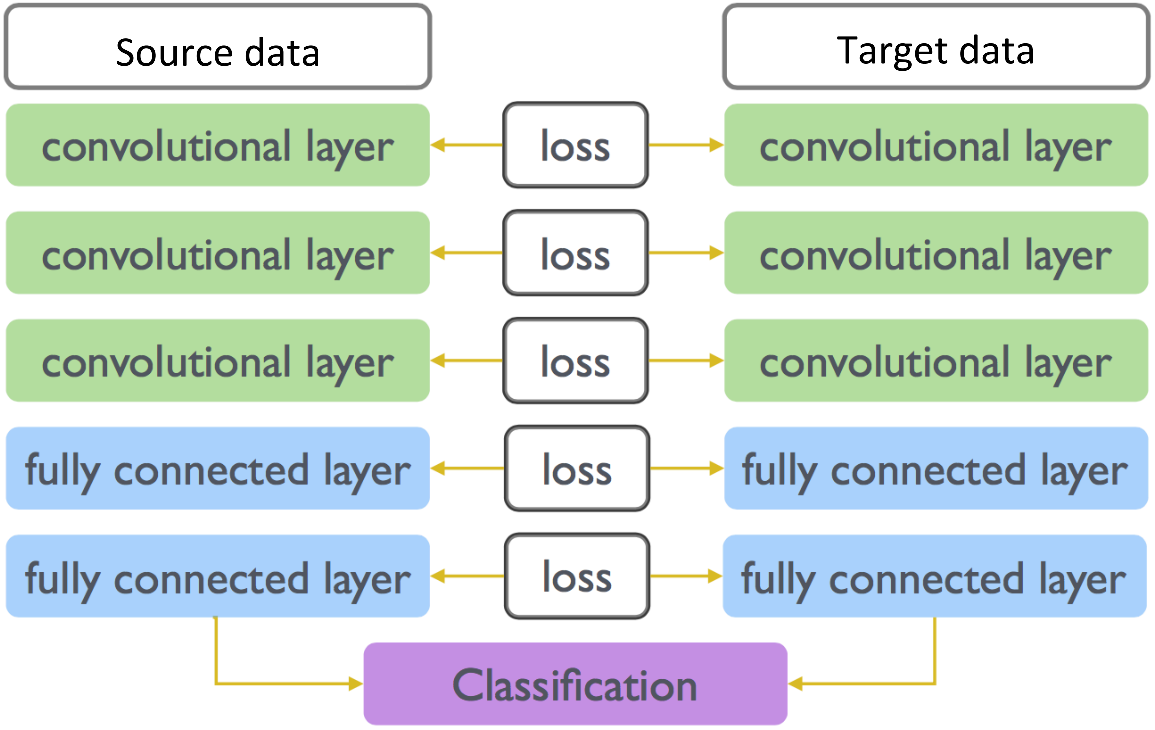
- Learn features that are invariant to the domain shift




- We believe that the domain shift should be modeled explicitly

# Beyond Sharing Weights

- We allow the weights to differ
- But regularize them to remain related



# Weight Loss

For each layer, we write

$$r_w(\theta_j^s, \theta_j^t) = \exp\left(\|a_j\theta_j^s + b_j - \theta_j^t\|^2\right) - 1$$

whose parameters we learn

# Complete Loss

$$L(\theta^s, \theta^t | \mathbf{X}^s, Y^s, \mathbf{X}^t, Y^t) = L_s + L_t + L_w + L_{MMD}$$

$L_s$ : Classification loss for the source samples

$L_t$ : Classification loss for the target samples

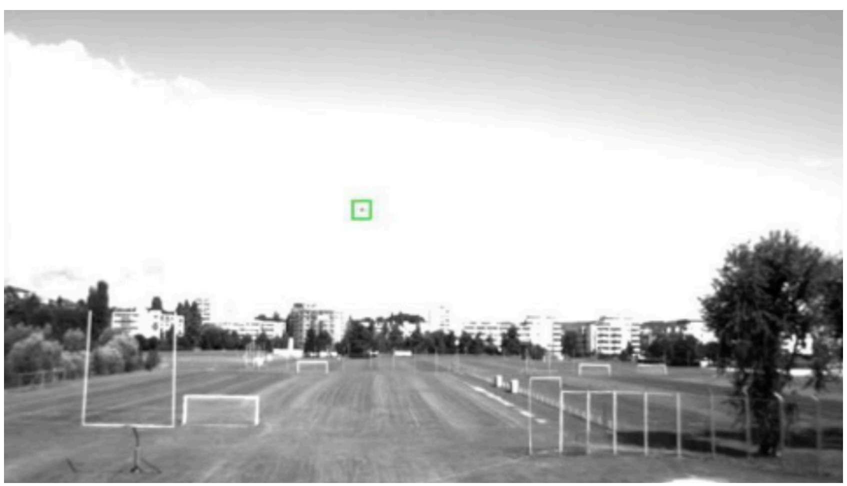
$L_w$ : Weight loss

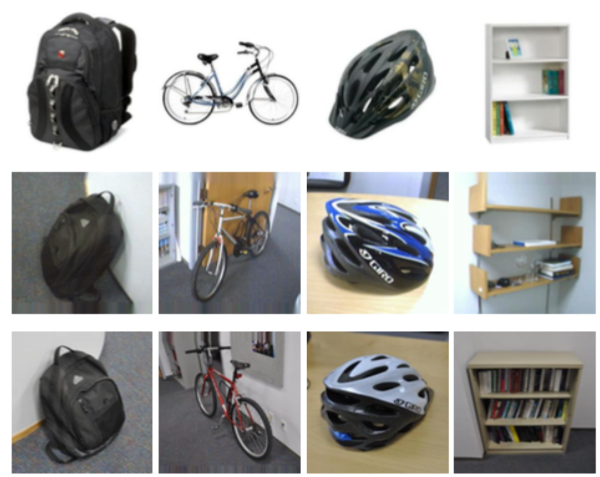
$L_{MMD}$: MMD loss for the source and target distributions

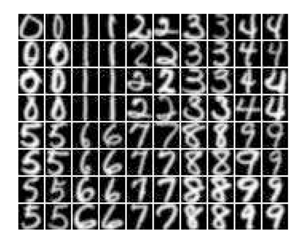
# Experiments

- Supervised and unsupervised domain adaptation

- Datasets:
  - UAV detection
  - Office 31
  - MNIST - USPS

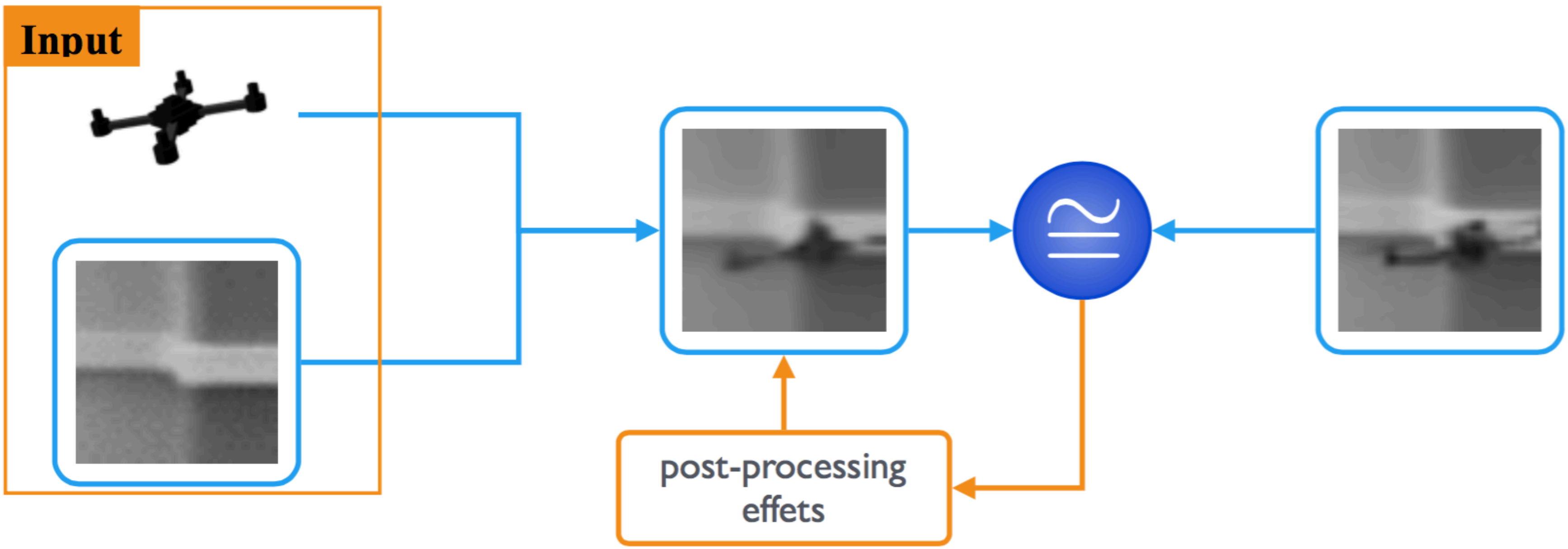






# UAV Detection

# Synthetic Data for UAV Detection



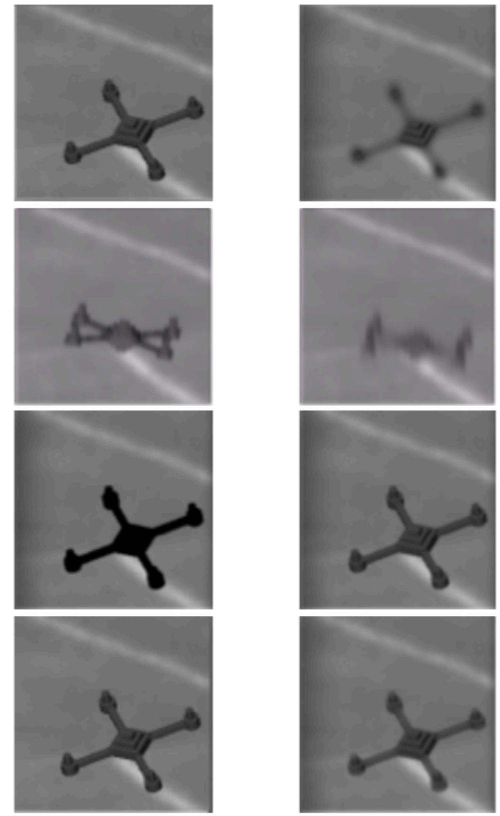
# Synthetic Data for UAV Detection

Boundaries blurring

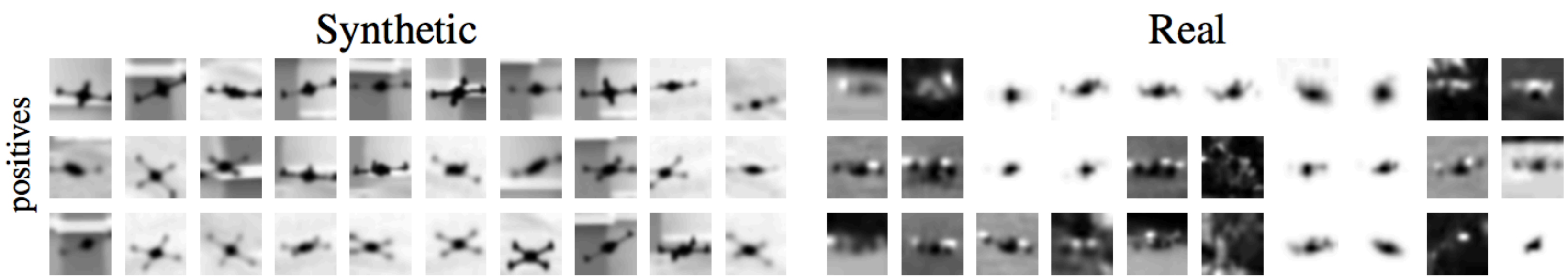
Motion blurring

Material properties

Gaussian noise

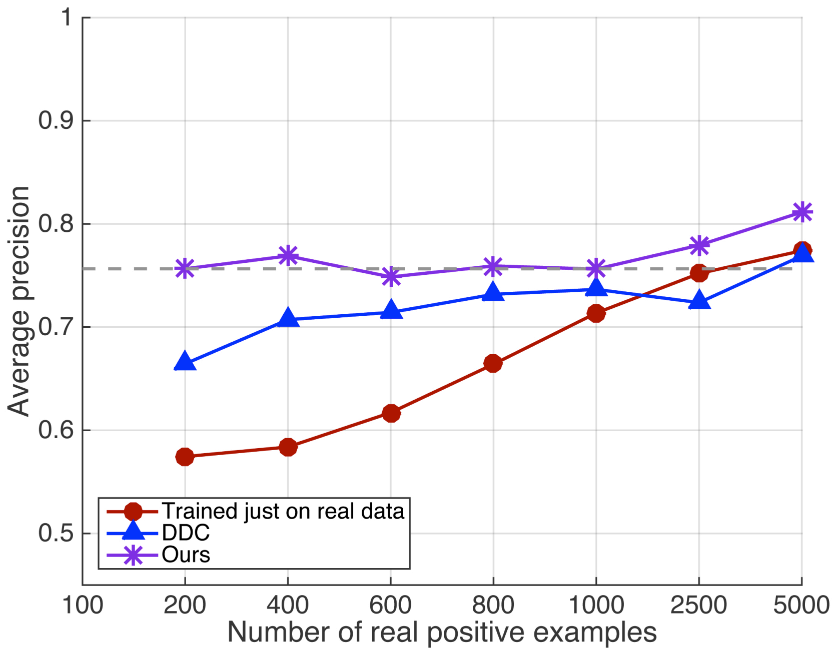


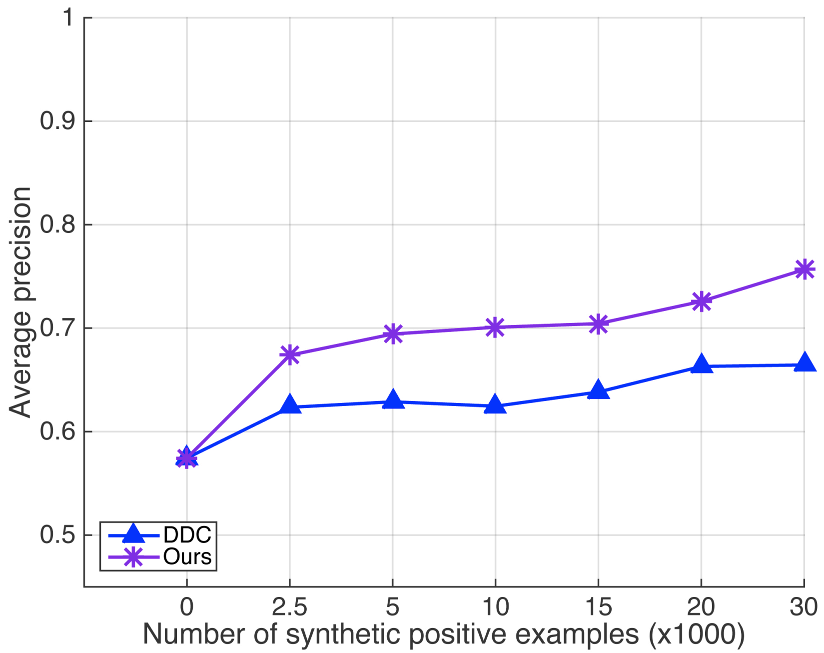
# UAV Detection: Results



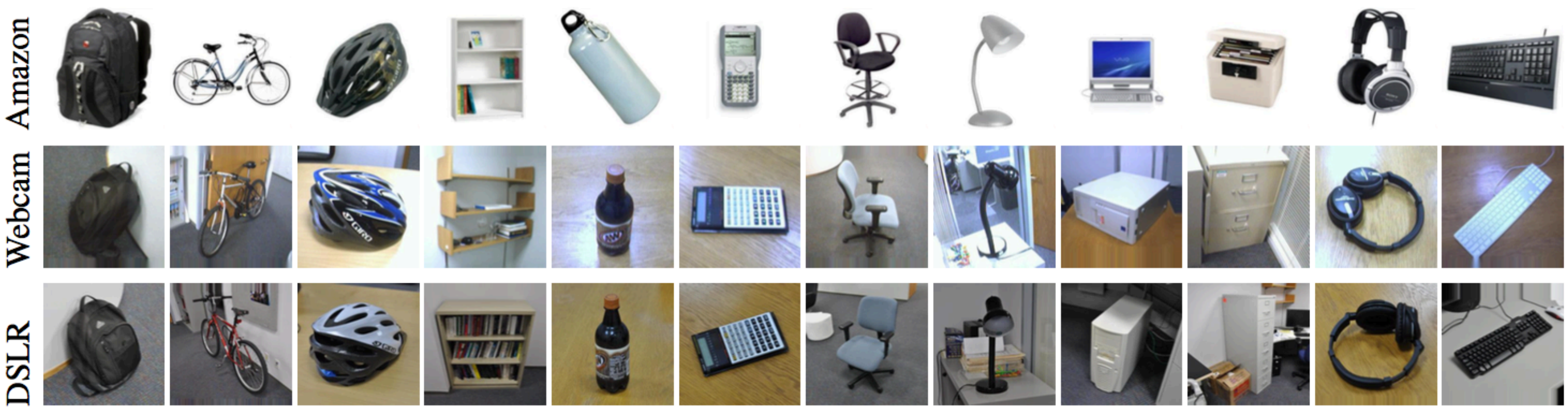

| | AveP (Average Precision) |
|---|---|
| *CNN (trained on Synthetic only (**S**))* | 0.314 |
| *CNN (trained on Real only (**R**))* | 0.575 |
| *CNN (pre-trained on **S** and fine-tuned on **R**):* | |
| *Loss:* $L_t$ | 0.612 |
| *Loss:* $L_t + L_w$ *(with fixed source CNN)* | 0.655 |
| *CNN (pre-trained on **S** and fine-tuned on **R** and **S**:)* | |
| *Loss:* $L_s + L_t$ [37] | 0.569 |
| DDC [7] *(pre-trained on **S** and fine-tuned on **R** and **S**)* | 0.664 |
| *Our approach (pre-trained on **S** and fine-tuned on **R** and **S**)* | |
| *Loss:* $L_s + L_t + L_w$ | 0.673 |
| *Loss:* $L_s + L_t + L_{MMD}$ | 0.711 |
| *Loss:* $L_s + L_t + L_w + L_{MMD}$ | **0.757** |

# UAV Detection: Influence of Synthetic Data



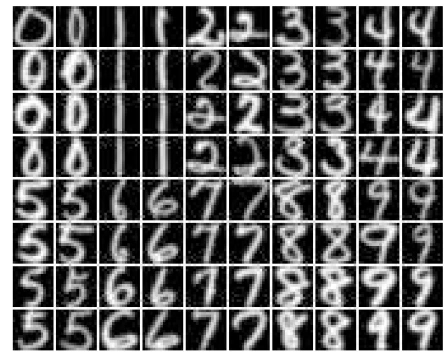



# Office 31




| | Accuracy | | | |
| --- | --- | --- | --- | --- |
| | A → W | D → W | W → D | Average |
| GFK [17] | 0.464 | 0.613 | 0.663 | 0.530 |
| SA [15] | 0.450 | 0.648 | 0.699 | 0.599 |
| DA-NBNN [41] | 0.528 | 0.766 | 0.762 | 0.685 |
| DLID [18] | 0.519 | 0.782 | 0.899 | 0.733 |
| DeCAF$_6$+T [29] | 0.807 | 0.948 | - | - |
| DaNN [27] | 0.536 | 0.712 | 0.835 | 0.694 |
| DDC [7] | 0.841 | **0.954** | 0.963 | 0.919 |
| Ours | **0.876** | 0.949 | **0.988** | **0.938** |

# MNIST - USPS

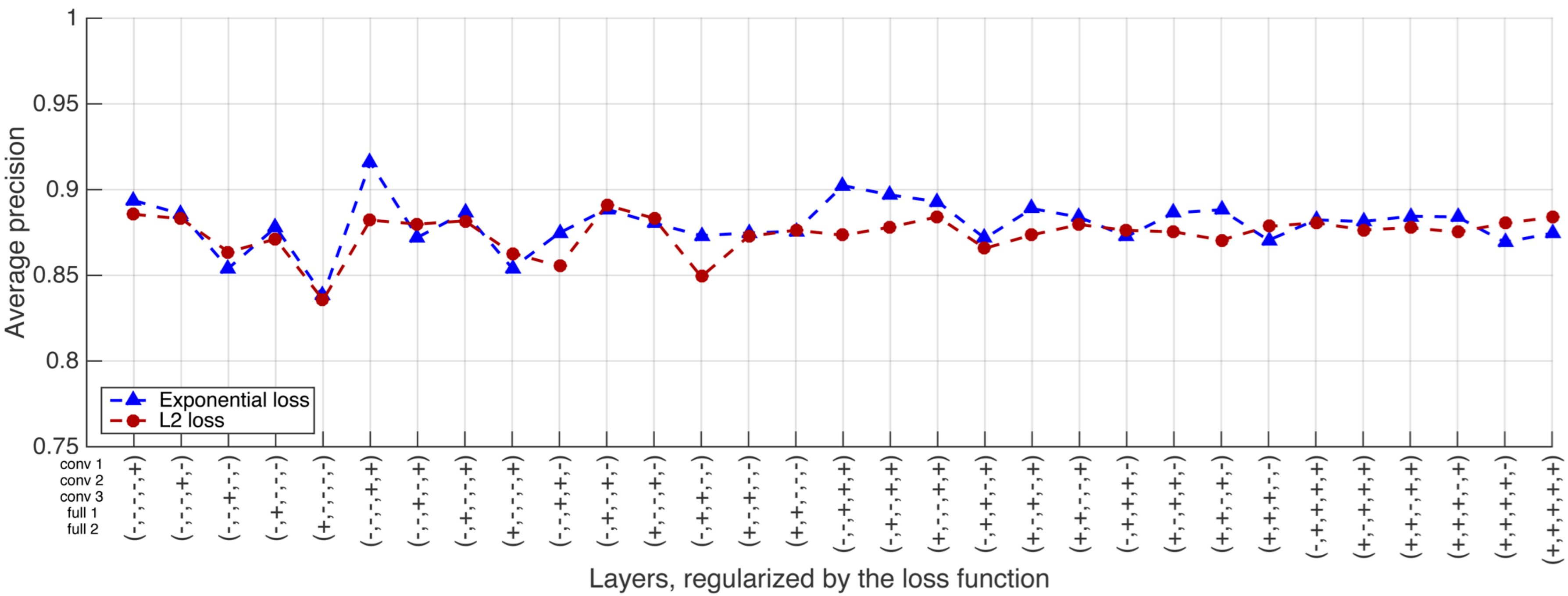


Accuracy

| method | NA | PCA | SA [15] | GFK [17] | TCA [44] | SSTCA [44] | TSL [45] | JCSL [43] | DDC [7] | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| M→U | 0.454 | 0.451 | 0.486 | 0.346 | 0.408 | 0.406 | 0.435 | 0.467 | 0.478 | **0.607** |
| U→M | 0.333 | 0.334 | 0.222 | 0.226 | 0.274 | 0.222 | 0.341 | 0.355 | 0.631 | **0.673** |
| AVG. | 0.394 | 0.392 | 0.354 | 0.286 | 0.341 | 0.314 | 0.388 | 0.411 | 0.554 | **0.640** |

# Network Design

- Not all layers should be allowed to have different weights
- The set of layers that should can be obtained by validation

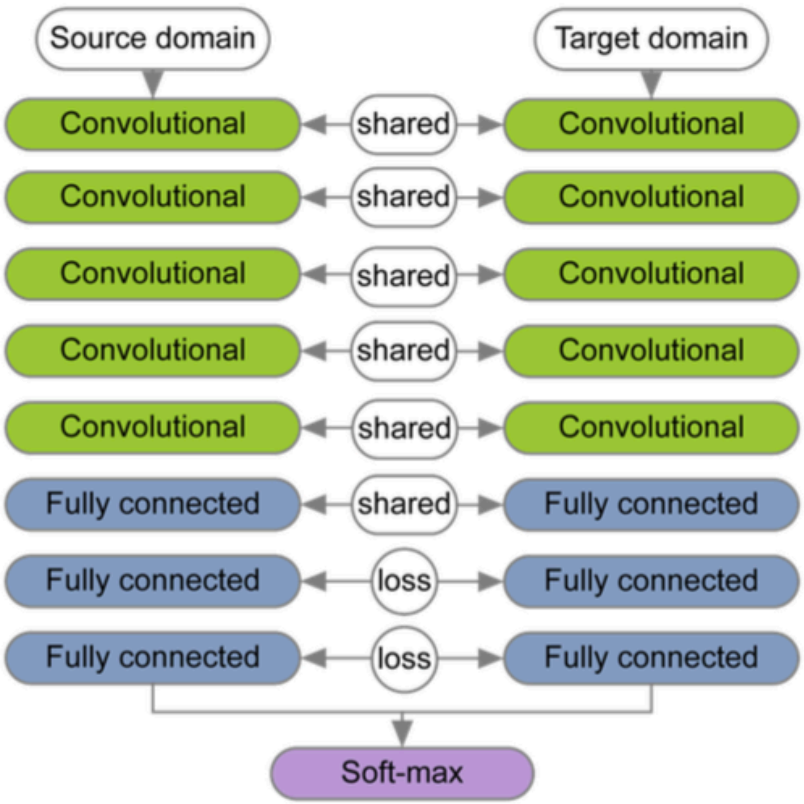



- For UAV detection, the first two convolution layers should adapt

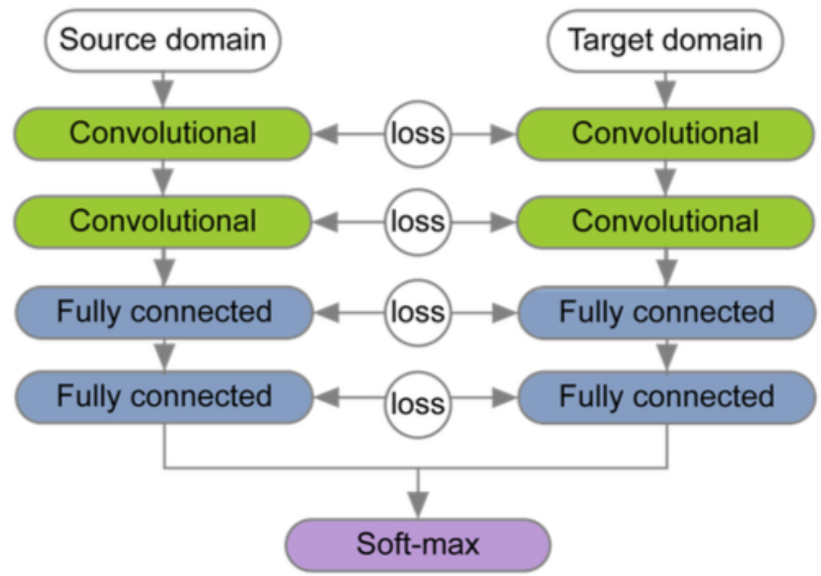
# Network Design

- This is problem dependent

Office 31 network




MNIST-USPS network



# Conclusions

- Deep Learning has lead to advances in domain adaptation
  - Many ideas used in the past can be translated to Deep Networks
- The weights should adapt to reflect the domain shift
  - Study more sophisticated weight transformations
  - Automatically learn which layers should be shared
- Synthetic data can help when real data is sparse
  - This can be interesting in other domains
  - Can we generate the data that is best suited to our purpose?