

# Deep Transfer Learning for Art Classification Problems

9<sup>th</sup> September 2018, Munich (GE)

**Matthia Sabatelli**<sup>1</sup>, Mike Kestemont<sup>2</sup>, Walter Daelemans<sup>3</sup>,  
Pierre Geurts<sup>1</sup>

<sup>1</sup>Montefiore Institute, Department of Electrical Engineering and Computer Science, Université de Liège, Belgium

<sup>2</sup>Antwerp Center for Digital Humanities and Literary Criticism (ACDC), Universiteit Antwerpen, Belgium

<sup>3</sup>CLiPS, Computational Linguistics Group, Universiteit Antwerpen, Belgium



*“Training a machine learning algorithm on a particular task (i.e. a classification problem) while using knowledge that the algorithm has already learned on a previously related task (i.e. a different classification problem)”*

We assume an input space  $\mathcal{X}_t$ , an output space  $\mathcal{Y}_t$ , and a probability distribution  $p_t(x, y)$  over  $\mathcal{X}_t \times \mathcal{Y}_t$

$$E_{(x,y) \sim p_t(x,y)} \{ \ell(y, f(x)) \}, \quad (1)$$

We want  $f : \mathcal{X}_t \rightarrow \mathcal{Y}_t$  where the only information available  $LS_t = \{(x_i, y_i) | i = 1, \dots, N_t\}$  drawn independently from  $p_t(x, y)$ .

In **Transfer Learning** we have an additional  $LS_s$  which differs from  $LS_t$  due to  $\mathcal{X}_s \neq \mathcal{X}_t$ ,  $\mathcal{Y}_s \neq \mathcal{Y}_t$ , or  $p_s \neq p_t$

We want to exploit  $LS_s$  together with  $LS_t$  to potentially minimize (1) better than when only  $LS_t$  is used for training

Table: An overview of the two datasets that are used in our experiments with  $N_t$  representing the amount of samples constituting the datasets and with  $Q_t$  the number of labels.

Challenge	Dataset	$N_t$	$Q_t$	% of overlap
Material	Rijksmuseum	110,668	206	None
	Antwerp	×	×	
Type	Rijksmuseum	112,012	1,054	$\approx 15\%$
	Antwerp	23,797	920	
Artist	Rijksmuseum	82,018	1,196	None
	Antwerp	18,656	903	

# ImageNet to the Rescue

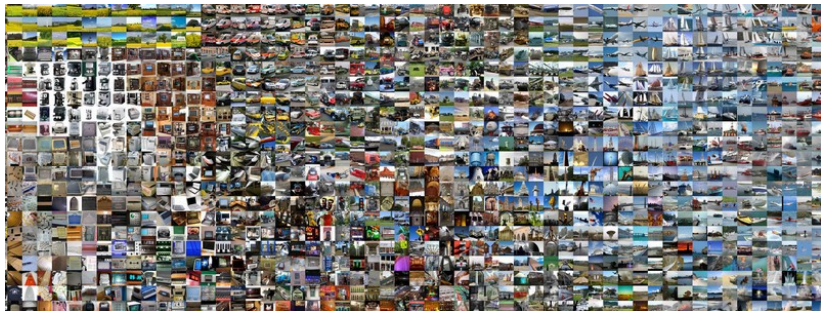


Figure: Courtesy of: <https://medium.com/@mozesr/script-to-get-images-from-image-net-org-7fe8592e6650>

# Deep Convolutional Neural Networks and TL Approaches

- Off the Shelf feature extraction
- Fine-Tuning
- Training from Scratch

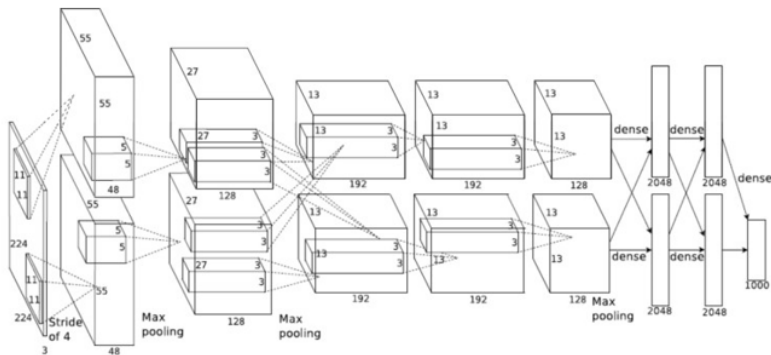


Figure: AlexNet; image taken from:  
<https://datascience.stackexchange.com>

# Neural Architectures

We use 4 Deep Convolutional Neural Networks:

- ☐ VGG19
- ☐ Inception V3
- ☐ ResNet
- ☐ Xception



# From Natural to Art Images: Material Classification

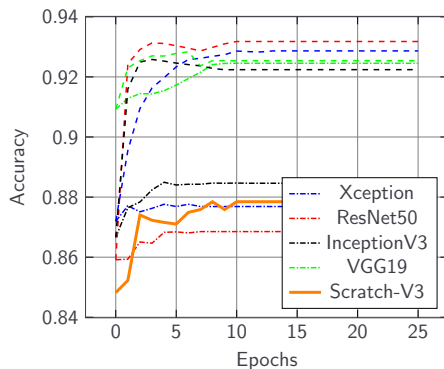


Figure: Comparison between the fine tuning approach versus the off the shelf one when classifying the material of the heritage objects of the Rijksmuseum dataset with respect to a DCNN trained from scratch.



# From Natural to Art Images: Type and Artists Classification

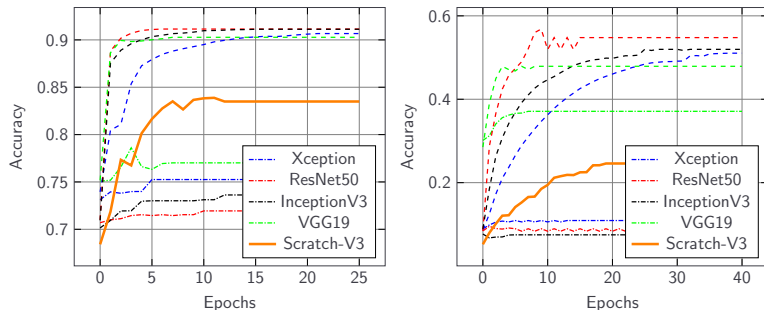


Figure: A similar analysis as the one which has been reported in the previous slide but for the second and third classification challenges (left and right figures respectively).

# From Natural to Art Images: Final Results

Table: An overview of the results obtained by the different DCNNs on the testing set when classifying the heritage objects of the Rijksmuseum. Bold results report the best performing architectures.

Challenge	DCNN	"off the shelf"	"fine-tuning"	Params	X
1	Xception	87.69%	92.13%	21K	2048
1	InceptionV3	88.24%	92.10%	22K	2048
1	ResNet50	86.81%	<b>92.95%</b>	24K	2048
1	VGG19	<b>92.12%</b>	92.23%	20K	512
2	Xception	74.80%	90.67%	23K	2048
2	InceptionV3	72.96%	91.03%	24K	2048
2	ResNet50	71.23%	<b>91.30%</b>	25K	2048
2	VGG19	<b>77.33%</b>	90.27%	20K	512
3	Xception	10.92%	51.43%	23K	2048
3	InceptionV3	.07%	<b>51.73%</b>	24K	2048
3	ResNet50	.08%	46.13%	26K	2048
3	VGG19	<b>38.11%</b>	44.98%	20K	512

- Significant improvements of fine tuning over the off the shelf approach
- The off the shelf approach still performs relatively well for the first 2 classification challenges (but is far in terms of performances when compared to fine tuning)
- Off the shelf classification fails when it comes to Artist Classification
- Benefits of an ImageNet initialization

# From One Art Collection to Another

- We use the fine-tuned DCNNs from ImageNet  $\Rightarrow$  Rijksmuseum ( $\hat{\theta}$ )
- We again use DCNNs trained on ImageNet only ( $\theta$ )
- We perform one more comparison: Off the Shelf vs Fine Tuning on the Antwerp Dataset



# From One Art Collection to Another: Results

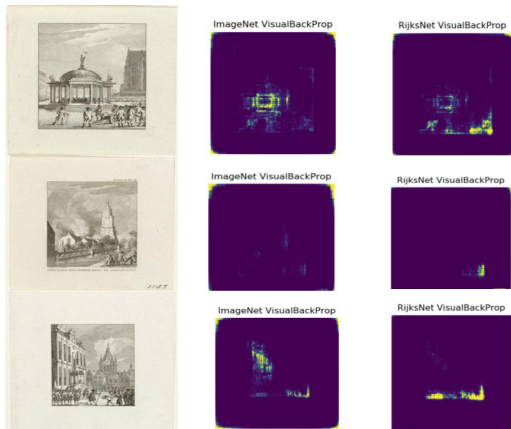
- The DCNNs which have been fine tuned on the Rijksmuseum outperform ImageNet's ones.
- Both if the off the shelf approach is used as if they are fine tuned
- Yield better generalization performances even on completely unseen labels → Our models a new better alternative that can be used in the field!

Challenge	DCNN	$\theta$ + off the shelf	$\hat{\theta}$ + off the shelf	$\theta$ + fine tuning	$\hat{\theta}$ + fine tuning
2	Xception	42.01%	62.92%	69.74%	72.03%
2	InceptionV3	43.90%	57.65%	70.58%	71.88%
2	ResNet50	41.59%	<b>64.95%</b>	76.50%	<b>78.15%</b>
2	VGG19	38.36%	60.10%	70.37%	71.21%
3	Xception	48.52%	<b>54.81%</b>	58.15%	58.47%
3	InceptionV3	21.29%	53.41%	56.68%	57.84%
3	ResNet50	22.39%	31.38%	62.57%	<b>69.01%</b>
3	VGG19	49.90%	53.52%	54.90%	60.01%

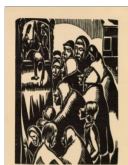
- The benefits of fine tuning are clear from the results
- Important to train DCNNs on similar  $LS_s \rightarrow LS_t$
- What changes between differently initialized DCNNs?
- **How can we interpret the classification performances of the DCNNs?**

# Selective Attention and Visual Backpropagation

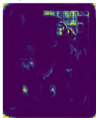
- Which pixel regions contribute the most to the softmax predictions of the DCNNs?
- Visual Backpropagation Algorithm



# Selective Attention Examples



ImageNet VisualBackProp



RijksNet VisualBackProp



ImageNet VisualBackProp



RijksNet VisualBackProp



ImageNet VisualBackProp



RijksNet VisualBackProp



ImageNet VisualBackProp



RijksNet VisualBackProp





- Extend these results to other smaller Datasets
- Investigate the use of Densely Connected Layers
- Combine the best parts of each explored architecture in a single one which will tackle all classification challenges at the same time

# Check out our GitHub Repository!

[https://github.com/paintception/  
Deep-Transfer-Learning-for-Art-Classification-Problems](https://github.com/paintception/Deep-Transfer-Learning-for-Art-Classification-Problems)

The screenshot shows the GitHub repository page for 'paintception / Deep-Transfer-Learning-for-Art-Classification-Problems'. At the top, there are navigation links for Code, Issues, Pull requests, Projects, Wiki, Insights, and Settings. Below these, a description states: 'Code and Manuscript for the ECCV 4th Workshop on Computer Vision for Art Analysis "Deep Transfer Learning for Art Classification Problems" paper.' There are tags for 'transfer-learning', 'computer-vision', 'deep-learning', 'digital-humanities', 'convolutional-neural-networks', and 'Manage topics'. The repository statistics show 34 commits, 1 branch, 0 releases, and 1 contributor. A table lists the files in the repository:

File	Description	Last Commit
figures	filter activations on images of paper	7 days ago
metadata	example of metadata file in .csv	5 days ago
models	type classification weights	7 days ago
paper	ECCV camera ready paper	5 days ago
saliency_maps_activations	proper path	7 days ago
transfer_learning_experiment	Update resnet.py	5 days ago
README.md	Update README.md	6 days ago

Below the file list, the 'README.md' file is expanded, showing the title 'Deep Transfer Learning for Art Classification Problems'. The text in the README states: 'We hereby release the code and models of the paper "Deep Transfer Learning for Art Classification Problems", which will be presented at the ECCV VisArt Workshop on Computer Vision for Art Analysis (September 2018, Munich GE). We investigate the performances of different Deep Convolutional Neural Networks (DCNNs) which pre-trained on the ImageNet dataset, aim to tackle 3 different art classification problems.'