

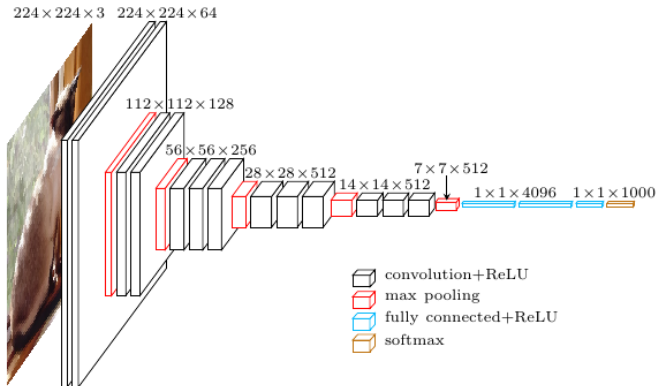
Neural Networks and Deep Learning: Transfer Learning

Nicolas Thome

Conservatoire National des Arts et Métiers (Cnam)
Département Informatique

Deep Features (DF) and Transfer Learning for Classification

- Visual recognition tasks: all performance re-benchmarked with DF since 2012

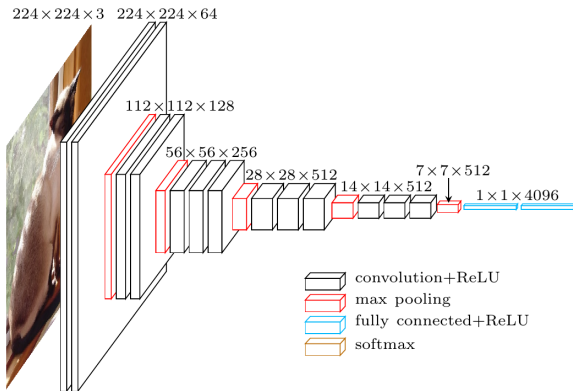


- Classification**: adapting final layer to match target classes

Deep Features (DF) and Transfer Learning for Classification

► Classification: adapting final layer to match target classes

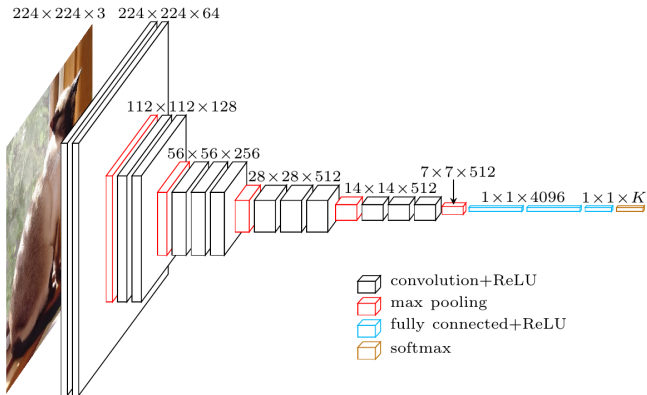
1. Remove last layer \Leftrightarrow ImageNet classes



Deep Features (DF) and Transfer Learning for Classification

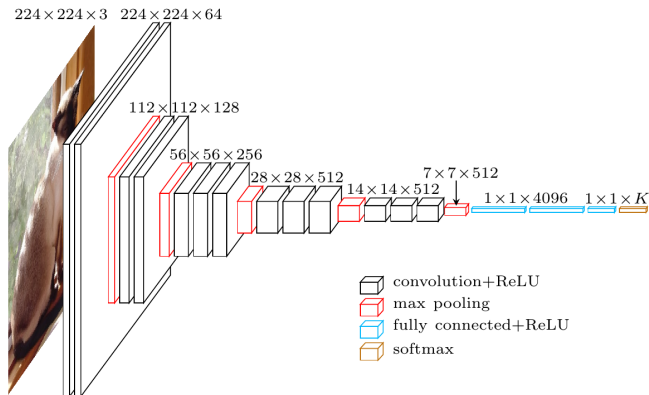
► Classification: adapting final layer to match target classes

2. Add transfer layer with K target classes, e.g. $K = 20$ VOC'07

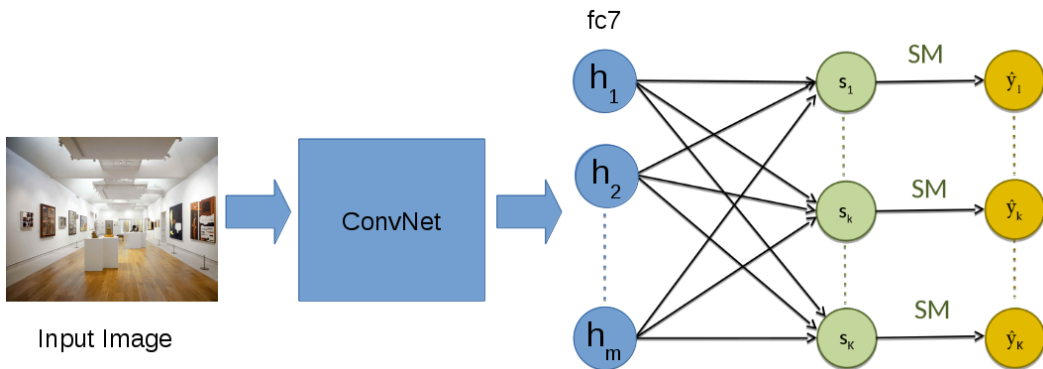


Deep Features (DF) and Transfer Learning for Classification

- ▶ **Classification loss?**
- ▶ **Training strategy?**, depending on:
 - a) Volume of the target dataset
 - b) Semantic proximity wrt ImageNet

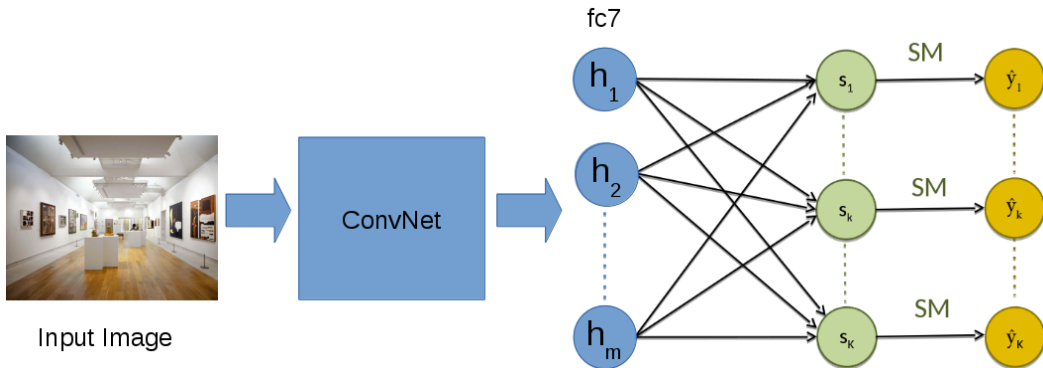


Transfer Learning for multi-class Classification



- ▶ Multi-class Classification: only one possible class per image, *i.e.* exclusive labels (e.g. ImageNet, MNIST, etc)
- ▶ Example here for transfer in MIT-67 (museum): indoor scene dataset

Transfer Learning for multi-class Classification



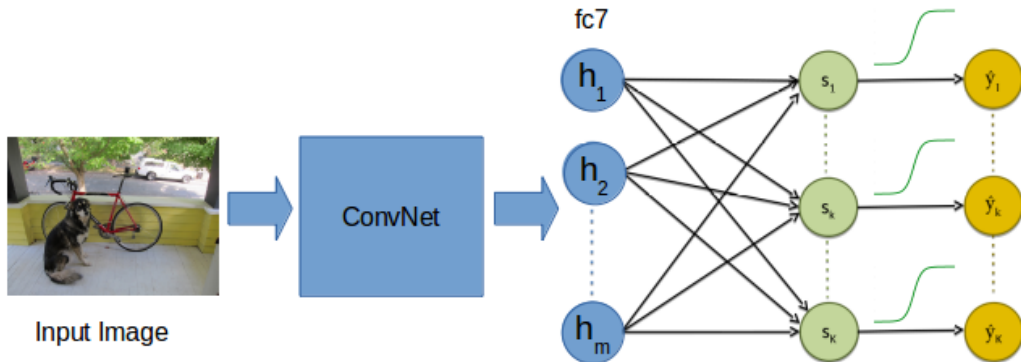
- ▶ Transfer layer of K classes + soft-max activation function
- ▶ $\mathbf{s}_i = \mathbf{x}_i \mathbf{W} + \mathbf{b}$, $\hat{y}_k \sim P(k/\mathbf{x}_i, \mathbf{W}, \mathbf{b}) = \frac{e^{s_k}}{\sum_{k'=1}^K e^{s_{k'}}}$
- ▶ Class parameters \mathbf{w}_k dependent

Multi-label Classification

- ▶ Several labels present for a given image, e.g. dog + bicycle + car (PASCAL VOC)
- ▶ Classification: train K binary models predicting class presence / absence

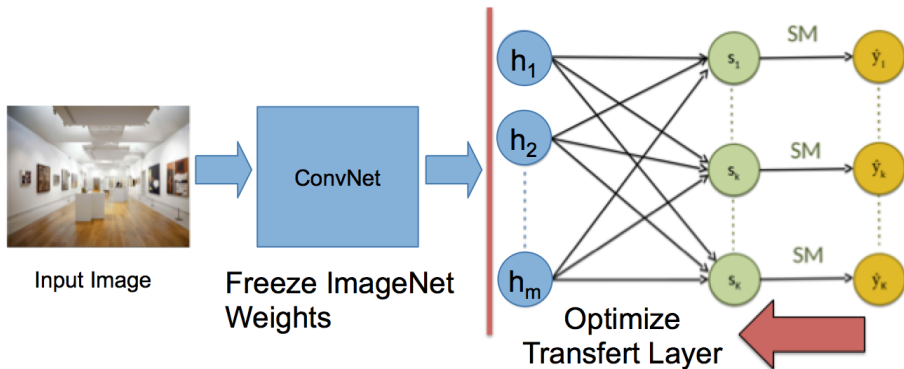


Transfer Learning for multi-label Classification

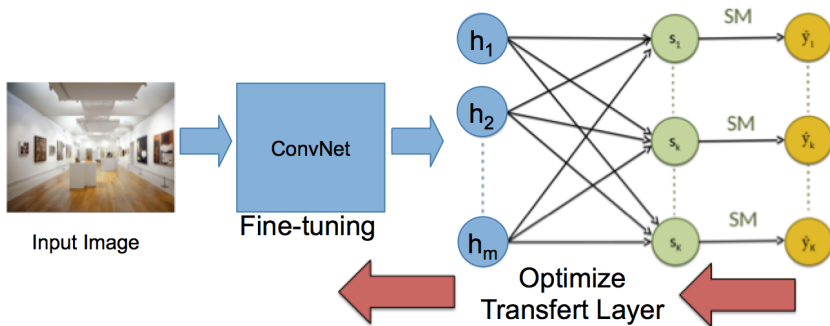


- ▶ Transfer layer of K classes + sigmoid activation function
- ▶ $\mathbf{s}_i = \mathbf{x}_i \mathbf{W} + \mathbf{b}$, $\hat{y}_k = \left[1 + e^{-\lambda \mathbf{s}_k} \right]^{-1}$
- ▶ Class parameters \mathbf{w}_k independent

Training Strategy: Pure Transfer



Training Strategy: Fine-Tuning



- ▶ Decreased learning rate for fine-tuned wrt from scratch parameters
e.g. factor 10 ↓

Training Strategy: Results

- ▶ **Small scale datasets** $[10^3, 10^4]$, **semantically close to ImageNet**
- ▶ Ex: VOC'07 (20 classes, 5000 ex)

Model	mAP (%)
VGG from scratch	≈ 40
Handcrafted FV	≈ 70
VGG pure transfer	≈ 83
VGG fine-tuning	≈ 85

- ▶ Fine Tuning > Transfer >> Handcrafted (BoW)
- ▶ From scratch low: not enough training data



Aeroplanes



Bicycles



Birds



Boats



Bottles



Buses



Cars



Cats



Chairs



Cows

Training Strategy: Results

- ▶ **Medium/large datasets $\geq 10^5$, semantically close to ImageNet**
- ▶ Ex: UPMC-food-101 (100 classes, 100000 ex)

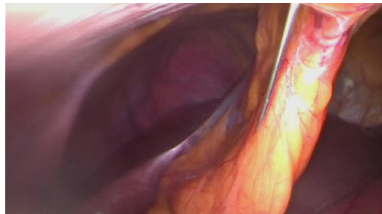


Model	mAP (%)
Handcrafted	≈ 25
VGG pure transfer	≈ 40
VGG from scratch	≈ 53
VGG fine-tuning	≈ 65

- ▶ **From scratch does work (well !)**
- ▶ **Fine Tuning >> From scratch >> Transfer >> Handcrafted**

Training Strategy: Results

- ▶ **Medium/large datasets $\geq 10^5$, semantically far from ImageNet**
- ▶ Ex: M2CAI'16 challenge: 22 videos, $\sim 10^5$ images, 8 classes



Model	Accuracy (%)
VGG pure tranfer	≈ 60
VGG from scratch	≈ 70
VGG fine-tuning	≈ 80

- ▶ **Fine Tuning >> From scratch >> Transfer**
- ▶ **Transfer already good baseline despite big visual content shift**

Transfer Learning: Conclusion

	Small visual shift	Large visual shift
Small dataset	Transfer top layers	Transfer lower layers
Larger dataset	Fine-tune top layers	Fine-tune all layers

- ▶ Small dataset, large visual shift: common in medical image classification
- ▶ **How implement transfer for localized tasks?**
⇒ following!

