# Conception d'architectures profondes pour l'interprétation de données visuelles

Taylor Mordan

**THÈSE DE DOCTORAT DE**
**SORBONNE UNIVERSITÉ**

Spécialité
**Informatique**
École Doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par


# Taylor MORDAN


Pour obtenir le grade de
**DOCTEUR de SORBONNE UNIVERSITÉ**

Sujet de la thèse :

# Designing Deep Architectures for Visual Understanding

**Conception d'architectures profondes pour l'interprétation de données visuelles**

soutenue le mardi 20 novembre 2018

devant le jury composé de :

| | | |
|---|---|---|
| M. Florent Perronnin | Naver Labs Europe | Rapporteur |
| M. Josef Sivic | INRIA – ENS | Rapporteur |
| M. Alexandre Alahi | EPFL – VITA Lab | Examinateur |
| Mme Natalia Neverova | Facebook AI Research | Examinatrice |
| M. Gilles Henaff | Thales LAS France S.A.S. | Encadrant (Invité) |
| M. Nicolas Thome | CNAM – CEDRIC | Co-directeur de thèse |
| M. Matthieu Cord | Sorbonne Université – LIP6 | Directeur de thèse |

# ABSTRACT

Nowadays, images are ubiquitous through the use of smartphones and social media. It then becomes necessary to have automatic means of processing them, in order to analyze and interpret the large amount of available data. In this thesis, we are interested in object detection, i.e. the problem of identifying and localizing all objects present in an image. This can be seen as a first step toward a complete visual understanding of scenes. It is tackled with deep convolutional neural networks, under the Deep Learning paradigm.

One drawback of this approach is the need for labeled data to learn from. Since precise annotations are time-consuming to produce, bigger datasets can be built with partial labels. We design global pooling functions to work with them and to recover latent information in two cases: learning spatially localized and part-based representations from image- and object-level supervisions respectively. We address the issue of efficiency in end-to-end learning of these representations by leveraging fully convolutional networks. Besides, exploiting additional annotations on available images can be an alternative to having more images, especially in the data-deficient regime. We formalize this problem as a specific kind of multi-task learning with a primary objective to focus on, and design a way to effectively learn from this auxiliary supervision under this framework.

# RÉSUMÉ

Aujourd'hui, les images sont omniprésentes à travers les smartphones et les réseaux sociaux. Il devient alors nécessaire d'avoir des moyens de traitement automatiques, afin d'analyser et d'interpréter les grandes quantités de données disponibles. Dans cette thèse, nous nous intéressons à la détection d'objets, i.e. au problème d'identification et de localisation de tous les objets présents dans une image. Cela peut être vu comme une première étape vers une interprétation complète des scènes. Nous l'abordons avec des réseaux de neurones profonds à convolutions, sous le paradigme de l'apprentissage profond.

Un inconvénient de cette approche est le besoin de données annotées pour l'apprentissage. Puisque les annotations précises sont longues à produire, des jeux de données plus gros peuvent être construits à l'aide d'annotations partielles. Nous concevons des fonctions d'agrégation globale pour travailler avec celles-ci et retrouver l'information latente dans deux cas : l'apprentissage de représentations spatialement localisée et par parties, à partir de supervisions aux niveaux de l'image et des objets respectivement. Nous traitons la question de l'efficacité dans l'apprentissage de bout en bout de ces représentations en tirant parti de réseaux complètement convolutionnels. En outre, l'exploitation d'annotations supplémentaires sur les images disponibles peut être une alternative à l'obtention de plus d'images, particulièrement quand il y a peu d'images. Nous formalisons ce problème comme un type spécifique d'apprentissage multi-tâche avec un objectif primaire, et concevons une méthode pour apprendre de cette supervision auxiliaire.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACRONYMS

| | |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| BoW | Bag-of-Words |
| ConvNet | Convolutional Neural Network |
| CRF | Conditional Random Field |
| CV | Computer Vision |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| DP-FCN | Deformable Part-based Fully Convolutional Network |
| DPM | Deformable Part-based Model |
| FCN | Fully Convolutional Network |
| FPN | Feature Pyramid Network |
| FV | Fisher Vector |
| GAFA | Google, Apple, Facebook and Amazon |
| GAN | Generative Adversarial Network |
| GAP | Global Average Pooling |
| GMM | Gaussian Mixture Model |
| GPU | Graphics Processing Unit |
| HOG | Histogram of Oriented Gradients |
| ILSVRC | ImageNet Large Scale Visual Recognition Challenge |
| IoU | Intersection over Union |
| LSVM | Latent Support Vector Machine |
| LUPI | Learning Under Privileged Information |
| mAP | mean Average Precision |
| MIL | Multiple-Instance Learning |
| mIoU | mean Intersection over Union |
| ML | Machine Learning |
| MTL | Multi-Task Learning |
| NMS | Non-Maximum Suppression |
| R-CNN | Region-based Convolutional Neural Network |

| | |
|---|---|
| R-FCN | Region-based Fully Convolutional Network |
| ReLU | Rectified Linear Unit |
| ROCK | Residual auxiliary blOCK |
| RoI | Region of Interest |
| SIFT | Scale-Invariant Feature Transform |
| SGD | Stochastic Gradient Descent |
| SPM | Spatial Pyramid Matching |
| SSD | Single Shot Detector |
| SSL | Semi-Supervised Learning |
| SVM | Support Vector Machine |
| TPU | Tensor Processing Unit |
| WILDCAT | Weakly supervIsed Learning of Deep Convolutional neurAl neTwork |
| WSL | Weakly Supervised Learning |
| YOLO | You Only Look Once |

CHAPTER 1

# INTRODUCTION

## Contents



## 1.1 Context

### 1.1.1 Scientific Context

Over the last few years, Artificial Intelligence (AI) has become a key technology in numerous domains and is today a major challenge to master in near future. Lots of companies (*e.g.* Google, Apple, Facebook and Amazon (GAFA)) carry research on this. Some examples of it already existing include dialogue systems, machine translation, speech recognition, autonomous driving, automatic analysis of social media content, games...

One important and active subfield of AI, in which we are interested in this thesis, is Computer Vision (CV), whose goal is to extract semantic information from images or videos, in order to understand their contents. CV already has a number of applications, *e.g.* robot navigation, content-based image retrieval, remote sensing, surveillance, medical imaging. Furthermore, with the increasing use of visual data on social media (*e.g.* Facebook, Instagram, Youtube), the need

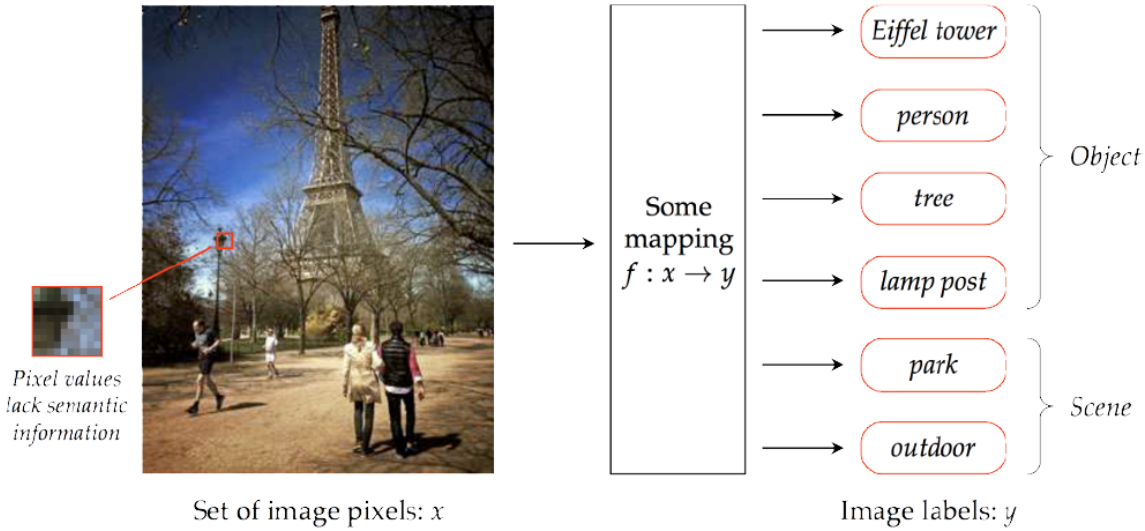



Figure 1.1 – **Illustration of CV tasks.** A model $f$ should produce semantic image labels $y$ from meaningless pixel values $x$, overcoming the semantic gap. (Credit: Hanlin Goh)

for automatic image analysis tools is more serious than ever, as all the annotation process, *e.g.* person detection and identification, action recognition, textual captioning for visually impaired users, sensitive content detection, cannot be carried out entirely by humans anymore. However, CV has inherent difficulties. In particular, the base element of visual data is the pixel, which does not carry any meaning. CV methods therefore need to abstract low-level pixels into high-level concepts, which is known as the semantic gap, illustrated in Figure 1.1. Moreover, images exhibit a large variability in many aspects, *e.g.* illumination, scale, point of view, and often depict unconstrained scenes, making the problem even harder.

At the heart of many successful AI applications, lies Machine Learning (ML). This subfield of AI is dealing with systems learning from data to improve themselves. Recently, ML has achieved great advances through Deep Learning (DL), which is a ML paradigm inspired by the brain. It aims at reproducing hierarchies of processing happening there, and at learning them directly from raw data. It is particularly suited to learn high-level, semantic concepts that are hard to exhaustively describe but that humans naturally understand, in contrary to rule-based systems or expert systems for instance, which are more tailored toward tasks where human reasoning can be fully explained. As such, DL seems particularly tailored to fill the semantic gap of CV.

A major drawback of DL is that it requires enormous quantities of data to learn from, which hinders it to work outside of specific applications for a few decades. Indeed, although DL dates back to the 1980s (Fukushima 1980; LeCun et al. 1989), it has achieved an astonishing success only recently. The reasons it has been possible are two-fold: the availability of large-scale datasets which have been

publicly released in the era of Big Data; and the advances of parallel computing, *e.g.* Graphics Processing Units (GPUs) or the newer Google's Tensor Processing Units (TPUs), required to process all these data in a reasonable time. The revolution of DL happened in 2012 in CV, on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015), a competition of image classification (*i.e.* identifying the natures of the main objects contained in pictures) shipping with an annotated dataset of more than 1.2 million images, which was orders of magnitude larger than any dataset released so far. The winning entry was a deep Convolutional Neural Network (ConvNet) named AlexNet (Krizhevsky et al. 2012), a DL approach that outperformed all other competitors by a significantly large margin. Since then, all the following winners have been using similar methods based on DL, and lots of other fields of AI, including most of CV tasks, have followed the path of DL.

After image classification, DL has progressively moved to other more complex CV tasks, in order to address the broader topic of scene understanding. In this thesis, we work on visual object detection, which also deals with the semantic recognition of objects, but with their spatial localization too. It is the first step toward a complete analysis of scenes, including detailed object description and identification of their relations.

### 1.1.2 Industrial Context

This thesis has been released in collaboration with Thales LAS France S.A.S., a company specialized in optronic systems for the defense area. It develops and builds imaging sensors working in various challenging conditions. Image processing and analysis are also a main part of their core business, with applications ranging from low- to high-level tasks.

The automatic detection and identification of targets or threats in images or videos is an important practical topic for Thales. This feature is needed in various products, and forms the core of several applications, such as surrounding monitoring or efficient analysis of vast areas. Some of them are time-critical and have to be performed quickly, at speed unmatched by humans, to trigger an appropriate response, while others would require large number of people and amount of time to be achieved in reasonable times. In all cases, automation of such tasks is beneficial both in terms of performances and resource constraints. DL is then an appealing approach that could have the potential to significantly improve the performances, as it has been shown in numerous public CV challenges.

Although visual object detection is a popular topic in academic scientific literature, applications targeted by Thales have particularities, due to being related to the defense domain. Especially, images can be hard to obtain in some applications due to confidentiality issues, or tedious and difficult to annotate because of associated constraints, *e.g.* lack of resolution. On the other hand, images are

often acquired by systems already providing other kinds of information about data, that can give some insights on their contents. For this reason, it is interesting for Thales to have learning approaches that can adapt to available annotations, depending on the contexts of the applications.

The methods developed in this thesis are in part inspired by the interests and goals of Thales, but are thought to be generic and applicable to generic visual object detection tasks and challenges, with the possibility to adapt them to any specific operational application.

## 1.2 Visual Understanding Framework

We now describe the framework used to learn most ML models, then move on to some common CV tasks we are interested in, image classification and object detection. Finally, we present some extension to the framework, with different hypotheses considered on the supervision available to learn from, namely Weakly Supervised Learning (WSL) and Multi-Task Learning (MTL).

### 1.2.1 Statistical Supervised Learning Framework

In a broad sense, ML deals with models improving at their tasks with experience. In statistical supervised learning, which will be used throughout this work, the goal is to approximate an unknown function $f^\star$, known through several annotated examples $(x, y^\star)$ only, with $x \in \mathcal{X}$ the input example and $y^\star = f^\star(x) \in \mathcal{Y}$ its corresponding correct label. All available examples are gathered in a dataset $\mathcal{D} = \left\{(x_i, y_i^\star)\right\}_{i=1}^N$ then defining the problem through $N$ examples. In CV, the input space $\mathcal{X}$ is often the set of all possible images (or video sequences) and the nature of the label space $\mathcal{Y}$ directly depends on the actual task.

Let us note $\mathcal{F}$ the space of functions from $\mathcal{X}$ to $\mathcal{Y}$. The learning problem is then to find the function $\hat{f} \in \mathcal{F}$ which best matches the dataset $\mathcal{D}$. For this, a loss function $\ell$ is commonly introduced to score predictions $y = f(x)$ of a model $f$ against the ground truths $y^\star = f^\star(x)$. We can then define the regularized empirical loss $\mathcal{L}$ of $f$, *i.e.* the loss over the whole dataset $\mathcal{D}$ with an additional regularization term $\mathcal{R}$ (Vapnik 1992):

$$\mathcal{L}(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(f(x_i), y_i^\star\right) + \mathcal{R}(f). \tag{1.1}$$

The regularization is used to encode *a priori* preference for specific properties, and is practically important to choose between several functions equally fitting the dataset, or to prevent overfitting, which happens when a model tries to perfectly fit the training examples beyond the meaningful variations of data,

resulting in it fitting the noise present in the dataset and not generalizing to other unseen data anymore. Both the loss function $\ell$ and the regularizer $\mathcal{R}$ have to be carefully selected to match the task and to encode meaningful information about the problem. The learning problem then reduces to the minimization of the regularized empirical loss to find the optimal model $\hat{f}$:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathcal{L}(f, \mathcal{D}). \tag{1.2}$$

The performances of the models are then often measured on a held-out annotated set, with a metric adapted to the problem and its requirements.

In practice, we consider parameterized functions $f_\theta \in \mathcal{F}$, where $\theta \in \Theta$ is the set of parameters, and optimization of Equation 1.2 is performed over parameters $\theta$. $\Theta$ is determined by the family of models chosen. Models with lots of parameters will easily fit complex datasets, but they will need more training data not to overfit them. In contrast, models with few parameters are simpler and have less capacity to represent data, but are easier to learn and more robust against overfitting. The size of the dataset $\mathcal{D}$ is therefore an important dimension to consider in any learning problem. If it contains more data, more complex models can be learned without overfitting, and finer aspects of the data can be understood. While in problems with small datasets, only the main variations of data can be modeled, by rather simple models with few parameters.

### 1.2.2 Image Classification

One of the first and most common tasks addressed by ML in CV is image classification, which aims at identifying the main object contained in an image within a predefined set of $C$ classes. In this case, $\mathcal{Y} = \{1, \ldots, C\}$ is the set of all $C$ classes. The metric commonly used to assess performance in image classification is accuracy, *i.e.* the fraction of images correctly recognized.

**Visual handcrafted representations.** Traditional approaches to image classification are two-stage pipelines: feature extraction and classification. The model $f$ is then written $f = g \circ \varphi$, with $\varphi$ a feature extractor and $g$ a classifier. Most often, the feature extractor $\varphi$ is fixed and only the classifier $g$ is learned. It is noticeable that the feature $\varphi(x)$ should encode all information relevant for the task from image $x$, while being invariant to any factor not affecting the class. Various classifiers $g$ can be used, *e.g.* k-nearest neighbors, logistic regression, Linear Discriminant Analysis, decision tree and forest, kernel methods (Hastie et al. 2001; Bishop 2006). In practice in CV, this is usually done with a linear Support Vector Machine (SVM) (Boser et al. 1992) for its effectiveness. It uses a regularization term aiming at maximizing the margin, *i.e.* the distance between the decision hyper-plane and

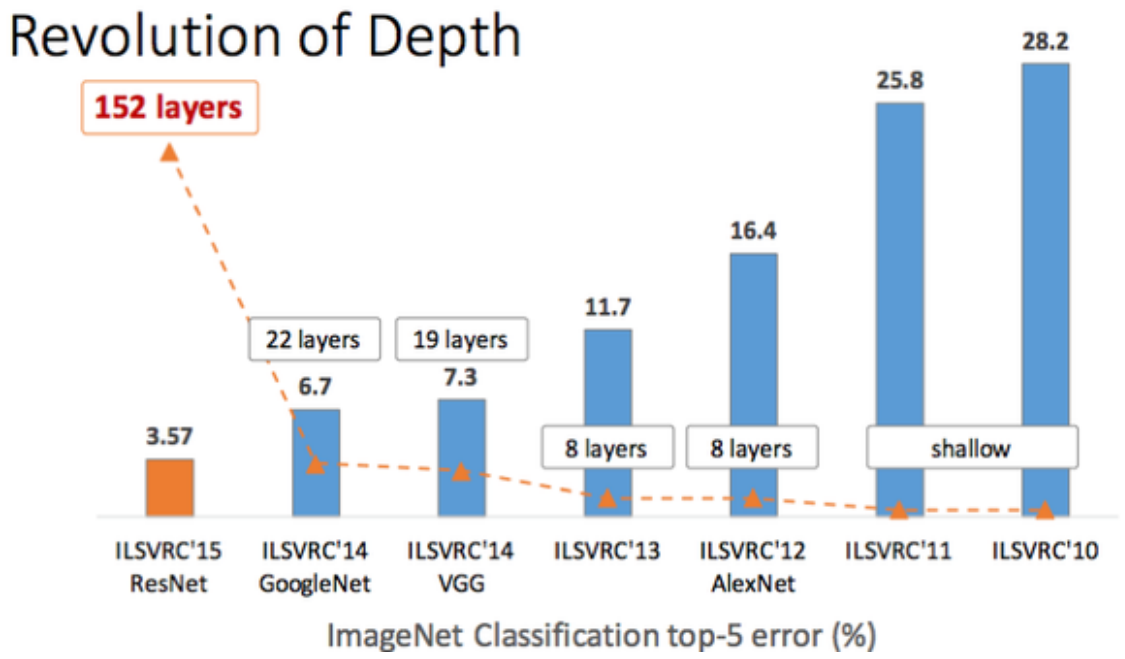



Figure 1.2 – **Evolution of results of ILSVRC** along with depths of winning networks. (Credit: Kaiming He)

the closest example. A larger margin then leads to a more robust classifier with better generalization (Vapnik 1995).

A common choice for feature extractor $\varphi$ is the Bag-of-Words (BoW) representation. It first appeared for textual information retrieval (Salton et al. 1986) and was later adapted to CV (Ma et al. 1999). The idea is to represent a document (or image) with a set of words (or local visual features) and their relative frequencies. As popularized by (Sivic et al. 2003), the common method in CV is to extract Scale-Invariant Feature Transform (SIFT) descriptors (Lowe 2004), as they are known to be rather robust to image transformations, but other descriptors are possible, *e.g.* Gabor filters (Fournier et al. 2001). Local features are then coded with k-means algorithm (Lloyd 1982) to obtain a dictionary, and words are pooled to compute histograms of occurrences.

A notable extension is Fisher Vector (FV) (Perronnin and Dance 2007; Perronnin, Sánchez, et al. 2010), which models the distribution of SIFT over the dataset with a Gaussian Mixture Model (GMM) and keeps first and second order statistics to have a finer description. FVs commonly use concatenation or Spatial Pyramid Matching (SPM) (Lazebnik et al. 2006) to keep coarse spatial information. Several variants exist, *e.g.* using different vectorial encodings (X. Zhou et al. 2010; Jégou et al. 2010; Picard et al. 2011) or a vectorial pooling (Avila et al. 2012).

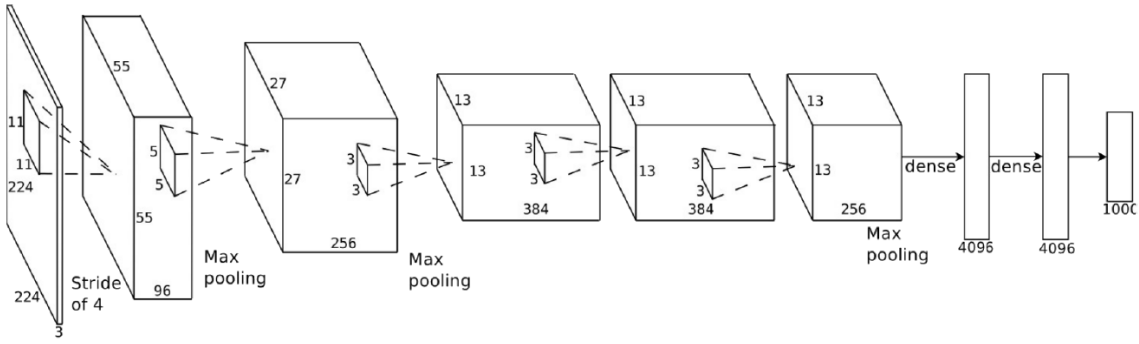
**Deep Convolutional Neural Networks.** Since the winning entry of AlexNet (Krizhevsky et al. 2012) at ILSVRC (Russakovsky et al. 2015) in 2012, state of the

| Network | Number of layers | Number of parameters |
|---|---|---|
| AlexNet (Krizhevsky et al. 2012) | 8 | 62M |
| VGG-16 (Simonyan et al. 2015) | 16 | 138M |
| Inception V1 (Szegedy, W. Liu, et al. 2015) | 22 | 6M |
| ResNet-50 (He, X. Zhang, et al. 2016a) | 50 | 26M |
| ResNet-101 (He, X. Zhang, et al. 2016a) | 101 | 45M |
| ResNet-152 (He, X. Zhang, et al. 2016a) | 152 | 60M |

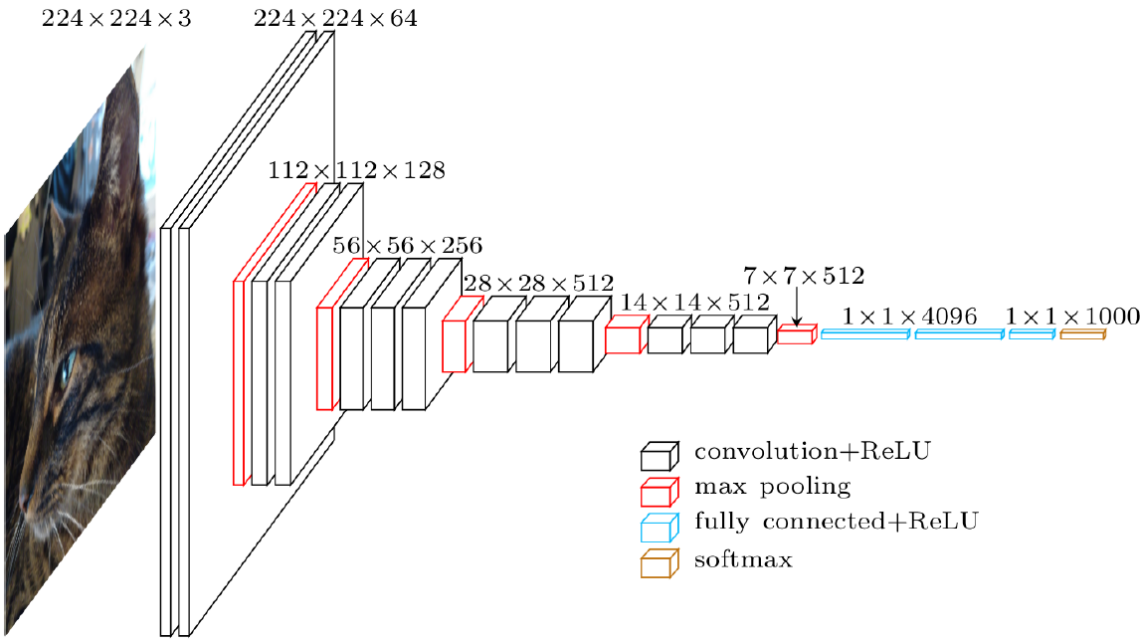
Table 1.1 – **Statistics of common deep ConvNet architectures** with depths and complexities (in parameters).

art in image classification is led by deep ConvNets, as shown in Figure 1.2. Starting with the formal neuron (McCulloch et al. 1943) and the Perceptron (Rosenblatt 1958), Artificial Neural Networks (ANNs) are a broad family of classifiers composed of neurons (*i.e.* computing units) interconnected through a directed graph. We are interested here in feedforward networks, which are based on directed acyclic graphs, *i.e.* the information flows in only one direction, from inputs to outputs. A remarkable property is that they can approximate any continuous function, if given enough neurons (Hornik 1991). Deep Neural Networks (DNNs) contain a large number of layers and can therefore learn complex mappings efficiently. They are often applied on raw data, on which they learn hierarchies of representations, *i.e.* multiple features at increasing levels of semantic, performing both feature extraction and classification jointly. In this case, the model $f$ of a network with $l$ layers writes as the composition $f = f_1 \circ \ldots \circ f_l$ of $l$ functions , each corresponding to a network layer. ANNs are commonly learned by using Stochastic Gradient Descent (SGD) to minimize the loss function, with gradients computed by gradient backpropagation (LeCun et al. 1989), an efficient algorithm leveraging the chain rule.

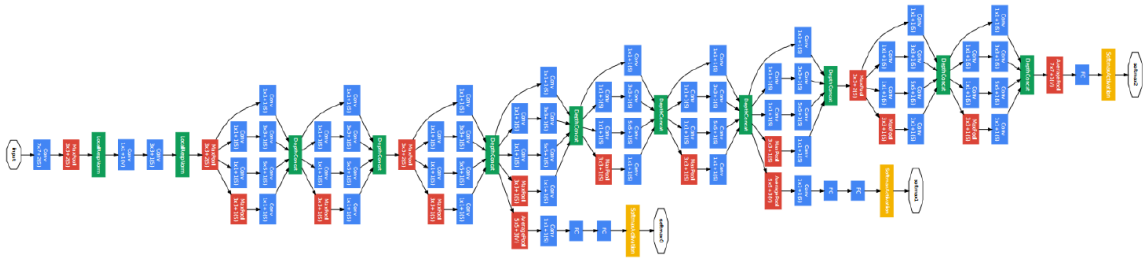
ConvNets (Fukushima 1980; LeCun et al. 1989; Krizhevsky et al. 2012) are a special kind of feedforward networks, which leverage convolution as their main operation. This is particularly suited to visual data as convolution takes the spatial coherence of pixels into account, and reduces the number of parameters by reusing local feature detectors over whole images, easing the training of networks. ConvNets are often built by stacking multiple convolutional layers, composed of a convolution operation whose filters are to be learned, a non-linear activation function, most often a Rectified Linear Unit (ReLU) (Krizhevsky et al. 2012), and eventually a pooling operation used to control the size of intermediate representations. Pooling is also used to build local invariance to transformations, similarly to what BoW or SPM do for full images. Since they contain a huge number of parameters, deep ConvNets had to wait until large-scale datasets, *e.g.* ImageNet



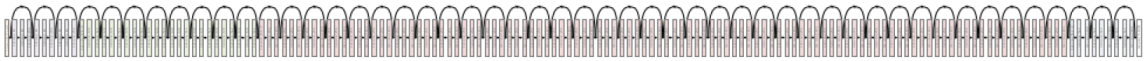

AlexNet (Credit: Krizhevsky et al. 2012)




VGG-16 (Credit: Simonyan et al. 2015)



Inception V1 (Credit: Szegedy, W. Liu, et al. 2015)



ResNet-152 (Credit: He, X. Zhang, et al. 2016a)

Figure 1.3 – **Common deep ConvNet architectures** used in CV.

(Russakovsky et al. 2015), and efficient hardware, *e.g.* GPUs, were available to be trained successfully, without overfitting, and in reasonable times. On the other hand, using this huge capacity, they produce powerful features that are learned to suit the problem. As such, these representations are more adapted than the BoW or FVs ones, and require less engineering.

The first modern architecture of deep ConvNet was AlexNet (Krizhevsky et al. 2012), depicted in Figure 1.3, which won ILSVRC 2012 with a large margin with respect to all other, shallow, methods. It was followed by similar improved versions, such as ZF Net (Zeiler et al. 2014) (winner of ILSVRC 2013) or VGG-S/M/F (Chatfield et al. 2014). The very deep architectures VGG-16/19 (Simonyan et al. 2015), showed that depth is a crucial property of networks. Increasing it with a simple structure improves performance, as observed in Figure 1.2, but results in a huge number of parameters, given in Table 1.1, and high inference time. VGG-16 is displayed in Figure 1.3. Successive Inception architectures (Szegedy, W. Liu, et al. 2015; Szegedy, Vanhoucke, et al. 2016; Szegedy, Ioffe, et al. 2017) designed efficient convolutional blocks to control complexity when increasing depth. The first version, which won ILSVRC 2014, is shown in Figure 1.3. However, increasing depth can lead to optimization issues. Several approaches solve this by structuring networks to ease gradients backpropagation, *e.g.* ResNet (He, X. Zhang, et al. 2016a; He, X. Zhang, et al. 2016b), ResNeXt (Xie et al. 2017), DenseNet (Huang et al. 2017). The resulting ResNet-152 won ILSVRC 2015 with a large margin, and is illustrated in Figure 1.3. Numerous approaches also try to reduce complexity of networks while keeping good results, *e.g.* SqueezeNet (Iandola et al. 2016), Xception (Chollet 2017), MobileNet (Howard et al. 2017; Sandler et al. 2018).

Deep ConvNets need huge amounts of labeled data to be trained properly, which might not be available for all tasks in sufficient quantities. However, the features learned on ImageNet (Russakovsky et al. 2015), have been shown to be generic and to transfer to other tasks, even outperforming previous handcrafted representations (Razavian et al. 2014; Azizpour, Razavian, et al. 2016). This makes possible to extend the success of deep ConvNets, and to apply them on other datasets, with fewer examples, by pre-training them on ImageNet and fine-tuning them on the target datasets, reducing the risk of overfitting due to small sample sizes.

### 1.2.3 Object Detection

We now turn to the task of object detection, which will be the main task considered in this work. It consists in both classifying (into one of the $C$ classes) and localizing (usually with a bounding box, *i.e.* with four coordinates) all objects in the images. The output domain for this task with any number of detections is then $\mathcal{Y} = \bigcup_{n=0}^{\infty} \mathcal{Y}_n$ where $\mathcal{Y}_n = \left(\{1, \dots, C\} \times [0,1]^4\right)^n$ is the set of $n$-tuple of detections, with normalized coordinates. Compared with image classification

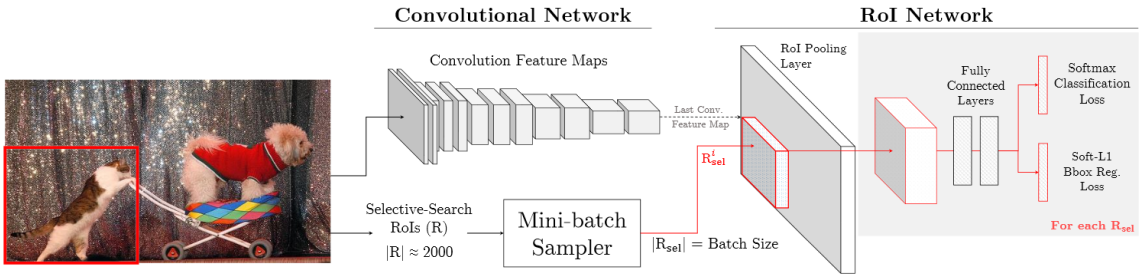



Figure 1.4 – **Fast R-CNN architecture.** It is composed of a ConvNet applied on whole images to produce global maps. Features are extracted within region proposals through a RoI pooling layer, and processed by two sibling sub-networks for classification and localization. (Credit: Abhinav Shrivastava et al. 2016)

task, it is more challenging as there can be an arbitrary large number of objects in images, while there is only one per image for classification, and it also requires to locate them. It is often evaluated with mean Average Precision (mAP) metric, the area under precision-recall curve, because positive (objects) and negative (background) examples are heavily unbalanced.

**Traditional object detectors.** The Viola-Jones face detector (Viola et al. 2001) was one of the first successful approach of ML to object detection. It relies on a large pool of simple Haar-like features, that are iteratively selected and turned into weak classifiers through AdaBoost (Schapire 1990). Inference speed is further improved by using an attentional cascade, where easy examples are removed early to focus computation with more Haar features on harder ones. To detect all faces, the classifier is applied in a sliding window fashion over images, *i.e.* at multiple overlapping positions in a multi-scale pyramid.

Following the feature extraction and classification framework, Histogram of Oriented Gradients (HOG) (Dalal et al. 2005) are computed in a sliding window way and locally normalized to produce discriminative features, that are then classified with a simple linear SVM classifier. Although being simple, this approach yields good detection results on human detection.

**Deep Convolutional Neural Networks.** As most other CV tasks, object detection is currently dominated by DL. It leverages powerful high-level visual features learned on ImageNet (Russakovsky et al. 2015) and transfers them to object detection by fine-tuning.

The first approach was DetectorNet (Szegedy, Toshev, et al. 2013) and uses a multi-scale coarse-to-fine mask regression ConvNet per class. DeepMultiBox (Erhan et al. 2014) exploits two networks to produce a fixed number of class-agnostic boxes, and to classify them to obtain the final set of detections. OverFeat

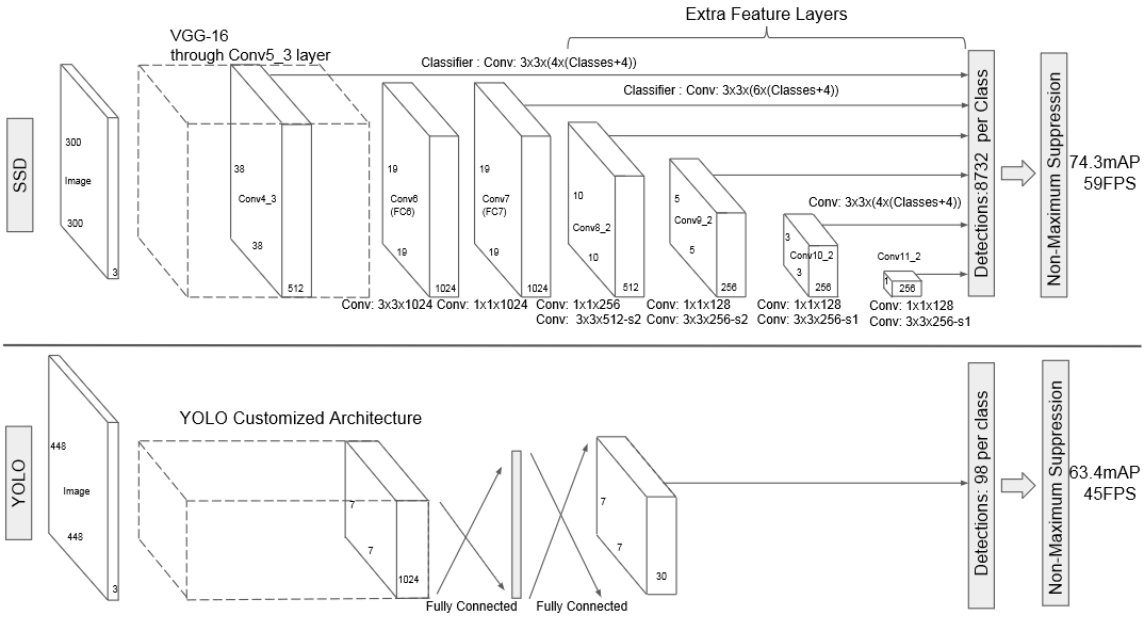



Figure 1.5 – **Examples of single-shot object detector architectures**: SSD (top) and YOLO (bottom). They do not compute region proposals, but detection is performed directly from global feature maps, with convolutions at multiple scales for SSD and with fully connected layers for YOLO. (Credit: W. Liu et al. 2016)

(Sermanet et al. 2014) applies a ConvNet in a sliding window fashion and uses two separate heads for classification and localization, learned separately.

The leading approaches in object detection are currently region-based deep ConvNets. Region-based Convolutional Neural Network (R-CNN) (Girshick, Donahue, et al. 2014) pioneered it by exploiting region proposal algorithms, *e.g.* Selective Search (Uijlings et al. 2013), Edge Boxes (Zitnick et al. 2014). Proposals are resized to a canonical size and forwarded through a ConvNet to extract deep learned features, then used for classification and localization with SVMs. Compared to sliding window, region proposals focus the computation on promising areas of images rather than evaluating the ConvNet everywhere, and balance positive and negative examples to ease learning by removing a large number of easy negative examples.

Following this way, several works improve on it. Fast R-CNN (Girshick 2015), depicted in Figure 1.4, significantly speeds up inference by sharing most of computation. Image-wise feature maps are computed and locally pooled within region proposals through a Region of Interest (RoI) pooling layer, leaving only a few RoI layers to be applied for each proposal separately (the RoI network on the right of Figure 1.4). In this configuration, most computations belong to the convolutional network and are shared between all regions from a same image. It also uses a multi-task setup to learn both classification and localization simultaneously

with the same network, with dedicated heads (*i.e.* prediction sub-networks) and losses for both tasks, as shown on the far right of Figure 1.4. Region-based Fully Convolutional Network (R-FCN) (Dai, Y. Li, et al. 2016) has almost no per-RoI computation by using a position-sensitive RoI pooling layer encoding accurate localization. Faster R-CNN (S. Ren et al. 2015) integrates the generation of region proposals into the ConvNet through an extended multi-task formulation, including additional network layers and losses for learning to produce them, rather than using an external algorithm. The multi-task approach is further exploited by Mask R-CNN (He, Gkioxari, et al. 2017), which adds another head for instance segmentation, improving detection performance.

There is also a second kind of object detectors, single-shot object detectors, which is not based on region proposals but directly predicting detections, *e.g.* You Only Look Once (YOLO) (Redmon, Divvala, et al. 2016; Redmon and Farhadi 2017), Single Shot Detector (SSD) (W. Liu et al. 2016), shown in Figure 1.5. Here, detection is performed directly on top of feature maps, globally for images with a convolution or a fully connected layer, to yield a set of detections of a predefined size. Since they do not rely on extracting region proposals, they are often lighter than region-based detectors, both running faster and containing less parameters, and are often the preferred methods for real-time inference. While their performances have long trailed behind those of region-based detectors, RetinaNet (T.-Y. Lin, Goyal, et al. 2017) has now closed the gap between the two kinds of approaches.

### 1.2.4 Weakly Supervised Learning

Despite excellent performances, deep ConvNets carry limited invariance properties, *i.e.* small shift invariance through pooling layers (Weng et al. 1992; Durand, Thome, et al. 2015; Shang et al. 2016; Blot et al. 2016). This yields issues when transferring networks pre-trained for image classification, *e.g.* on ImageNet where objects are centered and well identified, to other tasks, such as object detection, where objects can be in any numbers and at any location and scale. To optimally perform domain adaptation in this context, it becomes necessary to align informative image regions, *e.g.* by detecting objects (Oquab et al. 2015; Jaderberg et al. 2015), parts (N. Zhang, Donahue, et al. 2014; N. Zhang, Paluri, et al. 2014; B. Zhou, Khosla, et al. 2016; Kulkarni et al. 2016) or context (Gkioxari et al. 2015; Durand, Thome, et al. 2016). Although some works incorporate more precise annotations during training, *e.g.* bounding boxes (Oquab et al. 2014; Girshick, Donahue, et al. 2014), the increased annotation cost prevents its widespread use, especially for large datasets and pixel-wise labeling, *i.e.* segmentation masks (Bearman et al. 2016), as illustrated in Figure 1.6.

An option to gain strong invariance is to consider Weakly Supervised Learning (WSL), where image regions are explicitly aligned to learn localized latent

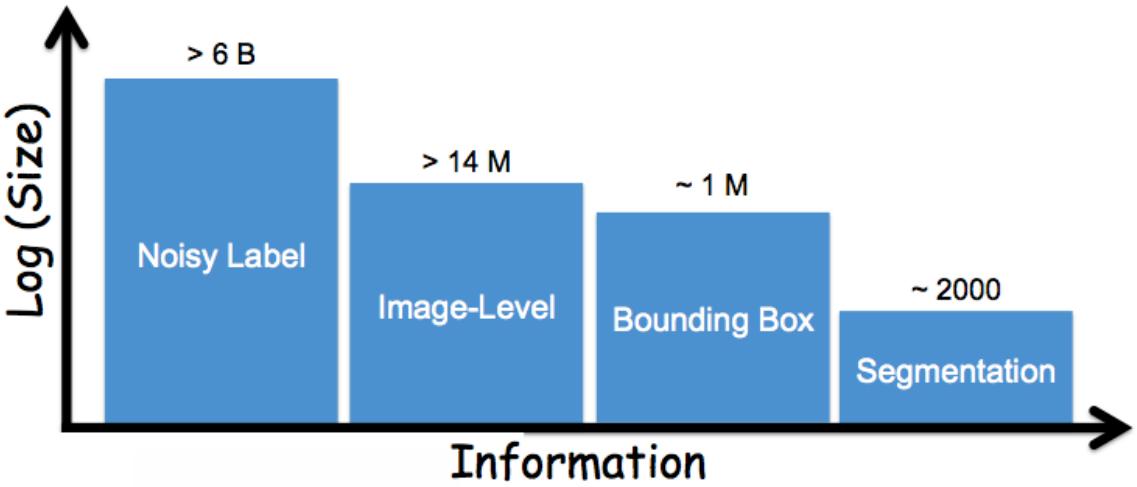



Figure 1.6 – **Amount of available annotations** depending on the precision required. Finer annotations take more time to be produced. (Credit: Pawan Kumar)

representations, better suited to transfer between tasks and datasets. WSL is a framework for learning problems where the available supervision contains only partial information about the expected outputs. In the general framework presented in Section 1.2.1, this would mean that labels are now written as $y^\star = (y_o^\star, y_h^\star)$ where only the observed label $y_o^\star$ is included in the dataset $\mathcal{D} = \left\{(x_i, y_{o,i}^\star)\right\}_{i=1}^N$, while the hidden part $y_h^\star$ would still be evaluated by the metric, and therefore need to be predicted. A common example is weakly supervised detection or segmentation, where the annotations are at image level only, *i.e.* presence or absence of object class in image, but results are to be given at instance or pixel level. WSL methods therefore provide ways to learn latent representations suited to the task from weaker supervision.

WSL problems are often dealt with through a Multiple-Instance Learning (MIL) framework (Dietterich et al. 1997) as it provides several ways of selecting regions and extracting relevant information. In MIL, an image is considered as a bag of instances (regions) with annotations at the bag level (*i.e.* image level) only, and the main issue concerns the aggregation function to pool instance scores into a global prediction, in order to learn from available supervision.

Several approaches use a MIL framework for image classification, with differences in handling positive and negative instances (Andrews et al. 2003; Felzenszwalb et al. 2010; Durand, Thome, et al. 2015; Durand, Thome, et al. 2018b). Different strategies have been explored to combine deep models and variants of MIL, with various region selection mechanisms (Oquab et al. 2015; W. Li et al. 2015; B. Zhou, Khosla, et al. 2016; Durand, Thome, et al. 2016)

MIL is also used in object detection by the seminal work of Deformable Part-based Model (DPM) (Felzenszwalb et al. 2010). It introduces a Latent Support

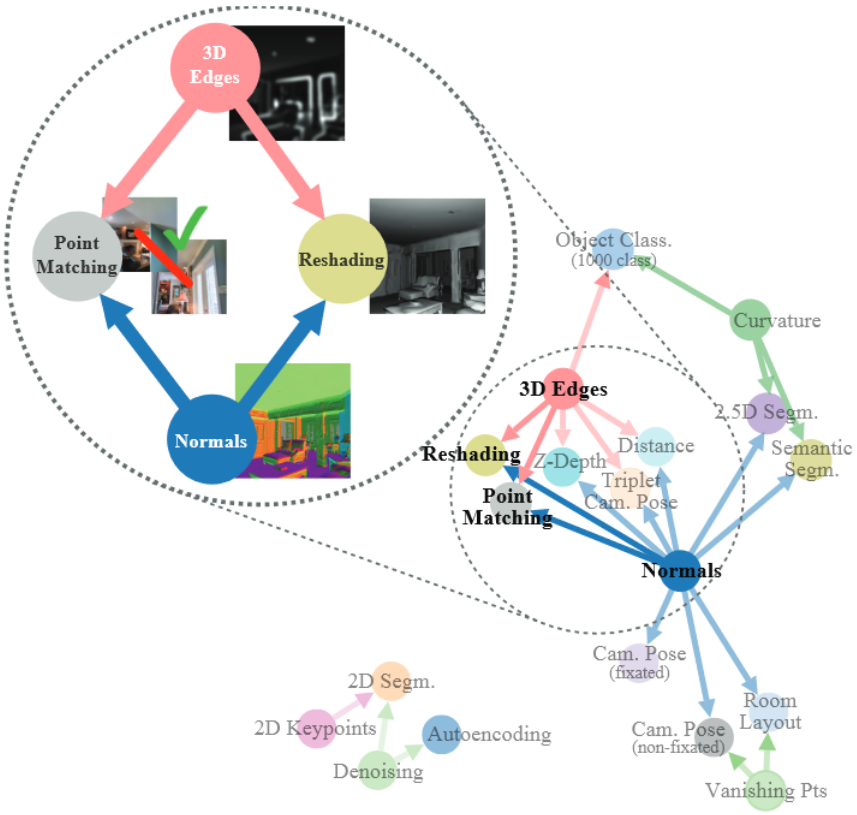



Figure 1.7 – **Synergies between tasks for transfer**, indicating how effective it is. (Credit: Zamir et al. 2018)

Vector Machine (LSVM) classifier to detect parts that are dynamically located. This part-based representation fits objects tightly and is especially useful to model non-rigid objects. This idea has also been introduced into deep models with direct generalizations of DPM to ConvNet (Girshick, Iandola, et al. 2015; Wan et al. 2015), although with limited success compared to other DL approaches.

### 1.2.5 Multi-Task Learning

As noted in Section 1.2.1, overfitting is a big issue in ML, especially when there are few training data. Existing solutions include using additional data or supervision. When it is of different nature, *e.g.* images from another modality or supervision for another task, it brings complementary information that can be exploited to regularize models.

The goal of Multi-Task Learning (MTL) (Caruana 1997) is to use a single model to perform multiple related but distinct tasks. It learns a shared representation that generalizes to all addressed tasks by leveraging different objectives and datasets. Intuitively, some tasks directly relate to each other, *e.g.* depth and surface normal estimation (Zamir et al. 2018), and so knowledge is solving one should help doing

so with the others by reusing learned insights. This is analyzed in Taskonomy (task-taxonomy, see Figure 1.7) (Zamir et al. 2018) to build a graph of amount of transfer between various common CV tasks, *i.e.* their synergies, in a principled way.

A MTL problem is defined as several learning tasks combined together, to be learned by a single model. The general framework from Section 1.2.1 can be slightly modified to suit MTL frame. Each example $x$ would be associated with a label $y_t^\star$ for each task $t$. The dataset then becomes $\mathcal{D} = \left\{(x_i, y_{1,i}^\star, \dots, y_{T,i}^\star)\right\}_{i=1}^{N}$ for $T$ tasks. Similar change needs to be applied to the loss function, with a dedicated one $\ell_t$ for each task, which are often summed, *i.e.* $\ell(y, y^\star) = \sum_{t=1}^{T} \ell_t(y_t, y_t^\star)$. It can be applied to multiple datasets, where each input is associated with a label for a given task, depending on the dataset the example comes from (Kokkinos 2017). It is also possible to use MTL on a single dataset annotated for multiple tasks, where ground truth labels are the concatenations of labels from all tasks (Zamir et al. 2018). In order to perform well, it is essential to encode hidden regularities and correlations between tasks, to benefit from synergy between them. This MTL strategy has been shown to improve results of individual tasks when learned together under certain conditions (Caruana 1997; Kokkinos 2017; Zamir et al. 2018). It also results in an efficient use of computational resources, where only one slightly bigger model is needed for all tasks simultaneously, compared to a different one used for each task separately. It is noticeable that this structure is employed in object detection, decomposed into two complementary tasks (object classification and bounding box localization) (Girshick 2015) or even more when also learning region proposal generation (S. Ren et al. 2015).

Although MTL is an old topic in ML, this is currently intensively revisited with DL. Deep MTL solutions consist in using different tasks and datasets and to share some intermediate representations of the networks between the tasks, which are optimized jointly (Bilen et al. 2016a; Kokkinos 2017; Meyerson et al. 2018). The crux of the matter is to define where and how to share parameters between tasks, depending on the applications. The common approach to MTL with DNNs is to have a feature extraction trunk shared between all tasks, with a dedicated prediction head for each task (Meyerson et al. 2018). However, it might not be optimal as different tasks often require features of different levels of abstraction, so the shared layers might be difficult to learn effectively. Some approaches focus on learning the optimal MTL architectures (Misra et al. 2016; Lu et al. 2017), while other explore relating every layer of the networks (Yang et al. 2017) or to relate layers at various depths to account for semantic variations between tasks (Meyerson et al. 2018). In UberNet (Kokkinos 2017), the goal is to learn a universal network which can share various low- and high-level dense prediction tasks, resulting in a very generic representation of images.

## 1.3 Motivations

When designing handcrafted features, they should be targeted to a particular task so that they encode all necessary information for it, while dropping all distracting factors. However, with the DL revolution that has happened over the last few years, features are no longer handcrafted but end-to-end learned by a network. There is not a total control over what features encode anymore, but this is left to be optimized by the model according to the learning objective, guided by the network and its structure. The focus has then shifted from feature design to network architecture design. A desired property of networks is to learn representations suited to the tasks at hand. This is achieved by integrating dedicated layers into the architecture, adapted to the level of available supervision, so that all relevant information is encoded by the model. This thesis studies three questions about network architecture design in order to improve extraction of latent information from various level of supervision:

**Global pooling function.** The main question in MIL and its variants is to define a global pooling function to convert all predictions at the instance level to a global one at the bag level. In WSL, the supervision is given at the image level only, *i.e.* presence or absence of classes in images, while the tasks might require predictions at the region level, *e.g.* object detection, or pixel-level, *e.g.* semantic segmentation. A structured pooling function considering the spatial nature of regions is then needed to learn localized representations that generalize to spatial tasks.

This question has been extensively studied in the context of deep ConvNets. The standard approach simply keeps the max-scoring region, *i.e.* the most informative one, for prediction (Oquab et al. 2015). It was extended to include several regions in the top instances model (W. Li et al. 2015) and with the Log-Sum-Exp pooling function (Sun et al. 2016), or all available regions with global average pooling (B. Zhou, Khosla, et al. 2016). Recent introduction of negative evidence (Parizi et al. 2015; Durand, Thome, et al. 2015; Durand, Thome, et al. 2016) has improved performances by also selecting min-scoring regions, which add evidence against some classes and help distinguish between similar classes. This is illustrated with an example in Figure 1.8.

The same question can also arise in object detection. This time, annotations are at the object level, *e.g.* with bounding boxes, so the learned representations are naturally localized within images. However, latent localization can be achieved within boxes, at the part level, in order to learn finer representations. Again, an aggregation function is required to score object regions (bag level) according to their parts (instance level), so that part localization can be learned.

This is the approach taken by DPM (Felzenszwalb et al. 2010), which learns to localize parts around objects so that the representations better fit them, especially for deformable objects where bounding boxes of fixed shape might be less adapted.

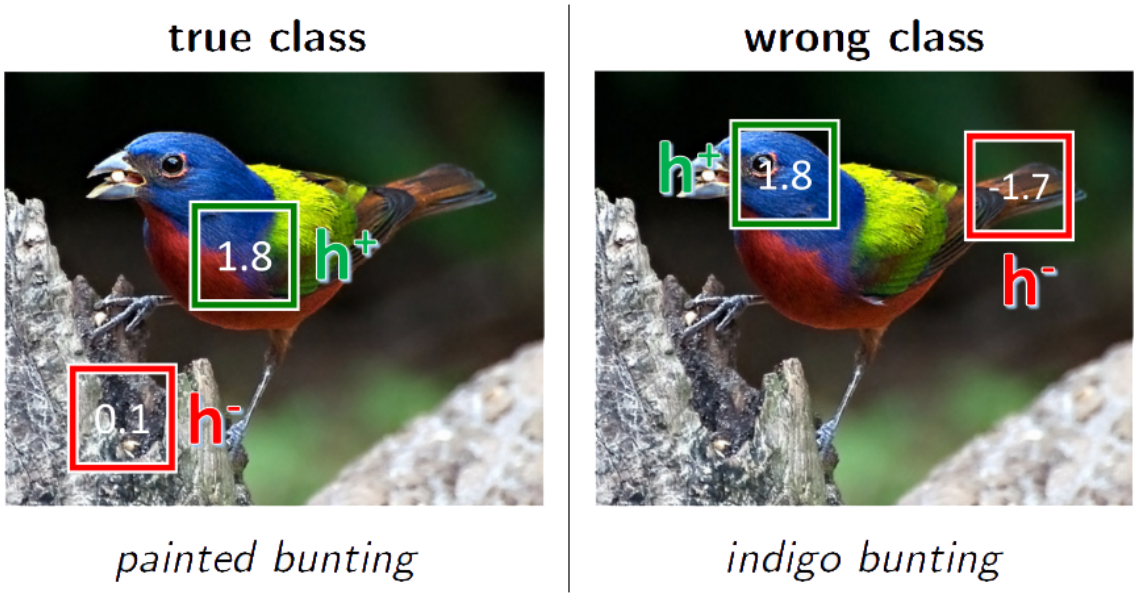



Figure 1.8 – **Illustration of negative evidence**. Both classes are birds with distinctive blue heads, which are detected well with high scores $h^+$. There is no evidence against the class *painted bunting* (the correct class), so the score $h^-$ is close to 0. However, the tail is in contradiction with the class *indigo bunting* (which is a bird fully blue), indicated by a highly negative score $h^-$. (Credit: Thibaut Durand)

Each class is represented by a root filter describing whole appearances of objects and a set of latent part filters to model local parts. These are localized around the root with a new LSVM, identifying the best position, *i.e.* the max-scoring one, for each filter among all possible locations. This is done in a latent way as there is no annotation regarding the parts. Final detection output then simply sums scores from root and all parts at selected locations. Most deep part-based networks are direct generalizations of DPM to DL (Girshick, Iandola, et al. 2015; Wan et al. 2015), using the same mechanism to select part location, but reformulated under the ConvNet structure.

Imposing structure between MIL instances is a way to enhance their labeling by taking their relative interactions into account. This leads to finer models able to express more relations between all instances, that describe data more accurately and yield increased performances. Several works have studied this path, by explicitly modeling relations between instances from a bag (Z.-H. Zhou et al. 2009) or from different bags (Deselaers et al. 2010). However, this has not been fully explored in the context of deep ConvNets and could yield benefits.

In this thesis, we are interested in improving global pooling functions in WSL by combining positive and negative evidences in a finer way, and by introducing structure into them.

**Efficient architecture for region computation.** When computing local features with deep models, the most naive approach is to rescale each region into a fixed-size vector adapted to the ConvNet architecture, as done in early works for detection, *e.g.* R-CNN (Girshick, Donahue, et al. 2014), or scene understanding (He, X. Zhang, et al. 2015; Gong et al. 2014; Oquab et al. 2014; Durand, Thome, et al. 2015).

Since this approach is highly inefficient, there have been extensive attempts for using convolutional layers to share feature computation. However, fully connected layers are beneficial in standard deep architectures, *e.g.* AlexNet (Krizhevsky et al. 2012) or VGG (Simonyan et al. 2015), but remove spatial information and so should not be applied on whole images. They are still used at the end of the networks in order to keep performances, but lead to a loss of efficiency. The common approach is to compute convolutional image-wise feature maps, to project image regions into feature regions, and to apply the fully connected layers to each region separately. This strategy has been employed in image classification with sliding window regions (Oquab et al. 2015; Durand, Thome, et al. 2016; B. Zhou, Khosla, et al. 2016), and object detection with region proposals (Girshick 2015; S. Ren et al. 2015).

Another issue for deep part-based networks (Girshick, Iandola, et al. 2015; Wan et al. 2015) is that the fully connected layers cannot neither be applied after part optimization in a straightforward manner, so they are then removed completely from the network. However, this did not yield significant improvement compared to the previous state of the art in object detection, contrary to what DPM achieved.

A second line of work in this thesis is the design of efficient convolutional architectures to allow end-to-end learning from image regions in WSL or object parts in detection.

**Learning with auxiliary supervision.** Adding more supervision should further boost performances, as observed by MTL (Caruana 1997), by allowing to learn other aspects of data, *e.g.* new modalities. It could especially be useful when there are few training examples, as it should regularize the training as more data are available to learn from. However, in the case where the additional supervisions are not as important as the main task, but only auxiliary in order to help training, care must be taken.

Common MTL approaches fail to acknowledge this difference between tasks, and give equal credits to all of them (Kokkinos 2017). This ends in optimizing average performance across tasks, while prioritizing one task might yield better results for this particular task but lower overall, as the capacities of models are bounded (Kokkinos 2017). Therefore, it requires a way to break symmetry between tasks during training.

Since additional auxiliary supervision is only relevant in order to train a better model, Learning Under Privileged Information (LUPI) (Vapnik and Vashist 2009)

seems to be an adapted framework. Here, some additional information is available only during training, but not at inference time. LUPI also naturally breaks the symmetry between tasks, according to the availability of their annotations. Several deep LUPI approaches have been proposed (Hoffman et al. 2016; Shi et al. 2017), but representations from standard and privileged supervisions often have light coupling. There is still need for a way to extract latent structure from all supervisions and align them to enhance the representation of the main task.

The last direction of research presented in this thesis deals with exploring ways to fully leverage additional supervision to benefit a given task, especially in the situations with few training data.

## 1.4 Contributions and Outline

In the following of this thesis, we propose and evaluate solutions to the limitations of current approaches identified before in Section 1.3. Throughout this thesis, we deal with increasing amounts of supervision, ranging from image-level annotations, to additional auxiliary supervision.

- Chapter 2: LEARNING LOCALIZED REPRESENTATIONS FROM IMAGE-LEVEL SUPERVISION
  MIL and its variants form an important framework for WSL. The main component is a global aggregation function allowing learning at the instance level from supervision at the bag level. Recently, negative evidence models have shown to bring complementary information to traditional MIL-based methods. We study ways to improve the pooling function with a better integration of negative evidence, as well as the introduction of structure between instances in order to learn richer models. In this chapter, we introduce WILDCAT, a model dedicated to learn localized representations from image-level supervision. For this, it uses a new global pooling function extending negative evidence models (Durand, Thome, et al. 2016) by differentiating contributions from positive and negative regions. It also builds structure with a multi-map transfer layer learning local features related to different class modalities in a weakly supervised way. This WSL strategy is integrated within a Fully Convolutional Network (FCN) to efficiently learn representations localized at the region level. WILDCAT is learned in an image classification setup, but can then be straightforwardly applied to WSL tasks, here pointwise object detection and semantic segmentation. This chapter is a joint work at equal contribution with Thibaut Durand.

- Chapter 3: LEARNING PART-BASED REPRESENTATIONS FROM OBJECT-LEVEL SUPERVISION
  Part-based representations are effective at modeling objects, especially de-

formable ones. However, most deep approaches using them either are learned in a strongly supervised setting, *i.e.* using part annotations, or do not fully exploit deep representations learned in ConvNets. We investigate the use of FCNs (Long et al. 2015) to perform MIL-based learning of deformable part-based representations within deep ConvNets at the object level, in order to learn fine representations from bounding box annotations. We introduce DP-FCN, a region-based object detector using latent deformable part-based representations to model objects and better fit them, especially when these are non-rigid. It exploits deformations for both detection sub-tasks: in object classification with a new MIL RoI pooling function identifying locations of parts, extending the ideas from DPM, and in bounding box regression with a deformation-aware localization module, leveraging computed deformations to better estimate shapes of objects. A further improvement of DP-FCN explicitly models interactions between all parts of a same object, with a Conditional Random Field (CRF)-based formulation for classification, and with a bilinear product for localization. DP-FCN also solves the issue of part computation efficiency by extending the R-FCN architecture (Dai, Y. Li, et al. 2016) based on a FCN.

- Chapter 4: LEARNING TASK-RELATED REPRESENTATIONS FROM AUXILIARY SUPERVISIONS
  Additional supervision is a way to increase performance and to reduce overfitting, especially with few training data. MTL seems an interesting framework to deal with this, but most approaches equally consider the additional data and the original ones, defining the main task. We tackle the problem of learning with auxiliary supervision, *i.e.* when results on the additional tasks are not the goal, and formalize it as a particular kind of MTL, named primary MTL, where the goal is the main task only. We introduce ROCK, a generic fusion block learning task-specific representations from various supervisions, to address it. ROCK is designed with two features tailored to primary MTL: it uses a residual connection between main and auxiliary tasks to make forward predictions explicitly impacted by the intermediate auxiliary representations, and incorporates intensive pooling operators in auxiliary sub-networks for maximizing complementarity of intermediate representations between tasks. It is applied to object detection with both geometric (depth and surface normal estimation) and semantic (scene classification) auxiliary tasks.

Lastly, Chapter 5 presents conclusions and discusses several directions for future work.

## 1.5 Related Publications

This thesis is based on the material published in the following papers:

- Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord (2017). "WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*;

- Taylor Mordan, Nicolas Thome, Matthieu Cord, and Gilles Henaff (2017). "Deformable Part-based Fully Convolutional Network for Object Detection". In: *Proceedings of the British Machine Vision Conference (BMVC)*, Best Science Paper Award;

- Taylor Mordan, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018a). "End-to-End Learning of Latent Deformable Part-Based Representations for Object Detection". In: *International Journal of Computer Vision (IJCV)*;

- Taylor Mordan, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018b). "Revisiting Multi-Task Learning with ROCK: a Deep Residual Auxiliary Block for Visual Detection". In: *Advances in Neural Information Processing Systems (NIPS)*.

CHAPTER 2

# LEARNING LOCALIZED REPRESENTATIONS FROM IMAGE-LEVEL SUPERVISION

## Contents




***Chapter abstract***

*In this chapter, we address the problem of learning a deep Convolutional Neural Network (ConvNet) for spatial tasks from image-level supervision only,* i.e. *exploiting less information than it is required to predict, a problem known as Weakly Supervised Learning (WSL). The common approach for WSL is to select some image regions to base the decisions on, rather than using entire images. This leads to localized representations that generalize better to spatial tasks. We propose to leverage a Fully Convolutional Network (FCN) to obtain region features efficiently, with computations shared within images. These features are then pooled to a single image-wise prediction, required by WSL to learn from image-level supervision. The global pooling function we use extends previous negative evidence models by differentiating positive and negative contributions, and additionally builds structure into the feature maps*

*by relating features to complementary class modalities. Our model, named WILDCAT, is learned for image classification, but the localized representations can directly be used for WSL tasks, here pointwise object localization and semantic segmentation.*

*The work in this chapter, at equal contribution with Thibaut Durand, has led to the publication of a conference paper:*

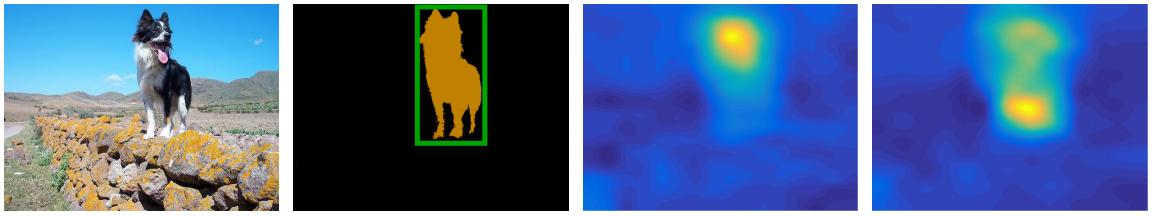
- Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord (2017). "WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

## 2.1 Introduction

Over the last few years, deep Convolutional Neural Networks (ConvNets) (Krizhevsky et al. 2012) have led a revolution in Computer Vision (CV), with outstanding results in most tasks. Since they need huge amounts of data to be trained on and generalize well, the common strategy is to leverage networks pre-trained on ImageNet (Russakovsky et al. 2015), containing more than a million of annotated images, and to fine-tune them on any target dataset and task. However, as already noted in Section 1.2.4, deep ConvNets only have limited invariance properties, which can make transfer hard. More precise annotations can be leveraged for this, but the annotation process can be tedious depending on the desired precision. As seen in Figure 1.6, finer annotations require more time to be produced, and so datasets with them contain fewer images. Another solution is Weakly Supervised Learning (WSL), already presented in Section 1.2.4, which we use in this chapter to learn localized representations from image-level labels only, *i.e.* presence or absence of classes in images. These representations, learned with image classification, are then directly used for WSL pointwise object localization and semantic segmentation.

Multiple-Instance Learning (MIL) (Dietterich et al. 1997) is an appealing framework for WSL, and there has been extensive work to generalize it, by questioning its assumptions. It has been adapted and further extended to Deep Learning (DL). In ConvNets, the main issue is to find a global pooling function that yields spatially localized features while learning from global labels only. It usually reduces to selecting the proper regions to base the predictions on, *e.g.* the most informative one only (Oquab et al. 2015) or all regions (B. Zhou, Khosla, et al. 2016).

The material presented in this chapter is a joint work with Thibaut Durand, and extends his previous researches on this topic. They are based on negative evidence framework (Parizi et al. 2015), a variant of MIL consisting in exploiting min-scoring



(a) image (b) predictions (c) *dog* heatmap 1 (d) *dog* heatmap 2

Figure 2.1 – **Example of WILDCAT localization and segmentation.** The predictions (b) are based on different class-specific modalities, here head (c) and legs (d) for the *dog* class.

regions as evidence against the presence of a class, through negative correlations. In particular, this acts as a regularization and can help distinguishing between classes with similar appearances or salient elements, by using other parts of objects or context, as already described in Figure 1.8. We now briefly describe two recent models, whose ideas are generalized in this chapter. MANTRA (Durand, Thome, et al. 2015) learns a Support Vector Machine (SVM) to classify pairs of latent variables, consisting of the min- and max-scoring image regions for a given class, and adds their scores to yield the final image score for this class. The prediction is then the class with the highest score. The symmetric nature of the formulation allows for efficient solving in several contexts. WELDON (Durand, Thome, et al. 2016) integrates this approach within a deep ConvNet and extends it to using multiple top and bottom instances (W. Li et al. 2015), *i.e.* the score of a class is the sum of the $k$ highest and $k$ lowest scores of regions for this class (with $k$ a hyper-parameter). Selecting multiple regions increases the robustness of the selection process, and helps dealing with images where evidence is distributed among several locations.

In this chapter, we present Weakly supervIsed Learning of Deep Convolutional neurAl neTwork (WILDCAT), a deep WSL model designed to learn spatial tasks from image-level annotations. Its main contribution deals with the global pooling function, needed in WSL to train with global labels, *i.e.* presence or absence of classes in images. It builds on the negative evidence framework presented above, and extends the aggregation strategy of WELDON by differentiating the contributions from positive and negative evidence terms. The proposed formulation generalizes multiple previous deep WSL models. To further localize learned features, so that they generalize better to spatial tasks, WILDCAT also builds structure into the representation by relating features to class modalities, *e.g.* head or legs for the *dog* class in Figure 2.1 (c) and (d), through a multi-map transfer layer. Once WILDCAT is trained for image classification with global labels, we apply it to two WSL tasks, pointwise object localization and semantic segmentation as illustrated in Figure 2.1 (b).

## 2.2 Related Work

To perform WSL, it is necessary to identify relevant information and to aggregate it to single values to learn from global labels. One option to detect informative image regions is to revisit the Bag-of-Words (BoW) model (Sivic et al. 2003), described in Section 1.2.2, by using deep features as local region activations (He, X. Zhang, et al. 2015; Gong et al. 2014; Goh et al. 2014) or by designing specific BoW layers, *e.g.* NetVLAD (Arandjelovic et al. 2016).

In the following, we discuss some approaches based on region selection.

**Global pooling function in deep ConvNets.** Different semantic categories are often characterized by multiple localized attributes corresponding to different class modalities (see for example head and legs for the *dog* class in Figure 2.1). Several strategies for deep models have then been explored to pool scores from regions of images to base decisions on informative elements (see Section 1.2.4).

Max pooling (Oquab et al. 2015) only selects the max-scoring region, as it should be the most discriminative for the task. However, this is prone to errors in the selection process, since only one region is kept, and all context is discarded. An alternative is to include all regions with Global Average Pooling (GAP) (B. Zhou, Khosla, et al. 2016), to bring robustness and take context into account, but this can also consider lots of uninformative regions. Intermediate approaches only select some relevant regions as a trade-off: Log-Sum-Exp pooling (Sun et al. 2016), Learning from Label Proportion (Felix Yu et al. 2013; Lai et al. 2014), and top instance (W. Li et al. 2015). Negative evidence models (Parizi et al. 2015; Durand, Thome, et al. 2015; Durand, Thome, et al. 2016) explicitly select regions accounting for the absence of the class. These regions bring contextual information to help differentiate between similar classes, *e.g.* with background or other parts of objects. However, there is not a method that is uniformly better than all others, the choice of the best approach depending on the task (Durand, Thome, et al. 2018a).

WILDCAT also exploits negative evidence, but differentiates positive and negative terms. Its pooling function generalizes most other common ones. We further base our model on a Fully Convolutional Network (FCN) backbone architecture (He, X. Zhang, et al. 2016a) to allow for efficient end-to-end training of the whole network for all image regions simultaneously.

**Other region selection mechanisms.** Global pooling functions are not the unique way of basing decisions on image regions. Similarly to WSL, attention-based models (K. Xu et al. 2015; Jaderberg et al. 2015; J. Zhang et al. 2016; H. Xu et al. 2016) select relevant regions to support decisions. However, WSL methods usually include some structure on the selection process while it is implicit in attention-based approaches.

Combining different regions has also been addressed through explicit context modeling (Gkioxari et al. 2015), or by modeling region correlations in RRSVM (Wei et al. 2016). For fine-grained recognition, multi-feature detection has been tackled in the fully supervised setting (H. Zhang et al. 2016; D. Lin et al. 2015; N. Zhang, Donahue, et al. 2014) and in WSL (Krause et al. 2014). In WILDCAT, we propose to build structure into representations through a multi-map transfer layer, which relates features to class modalities, as depicted in Figure 2.1.

**Weakly Supervised Learning for localization and segmentation.** Several recent works have applied WSL to the tasks of object localization and semantic segmentation.

WSL localization can be addressed with label co-occurrence information and a coarse-to-fine strategy based on deep feature maps to predict object locations (Bency et al. 2016). ProNet (Sun et al. 2016) uses a cascade of two networks: the first generates bounding boxes and the second classifies them. Similarly, WSDDN (Bilen et al. 2016b) proposes a specific architecture with two branches dedicated to classification and detection.

Many WSL segmentation methods are based on MIL framework: MIL-FCN (Pathak, Shelhamer, et al. 2015) extends it to multi-class segmentation, MIL-Base (Pinheiro et al. 2015) introduces a soft extension of it, EM-Adapt (Papandreou et al. 2015) includes an adaptive bias into the framework, and Constrained CNN (CCNN) (Pathak, Krahenbuhl, et al. 2015) uses a loss function optimized for any set of linear constraints on the output space.

WILDCAT is not specific to localization nor segmentation. It is learned for image classification and the localized representations obtained can be straightforwardly used for spatial tasks such as these two, with minor changes in the predictor to adapt it to the task at hand.

## 2.3 WILDCAT Model

The overall WILDCAT architecture is depicted in Figure 2.2. It is based on a FCN which is suitable for spatial predictions (Long et al. 2015), allowing end-to-end training for all image regions by sharing computations (Section 2.3.1). All regions are encoded into multiple class modalities with a WSL multi-map transfer layer to build structure into the representations (Section 2.3.2). Feature maps are then combined separately to yield class-specific heatmaps that can be globally pooled to get a single probability for each class, using a new spatial aggregation module. This generalizes previous global pooling functions, including negative evidence models, with positive and negative contributions computed from different numbers $k^+$ and $k^-$ of regions, and with an additional hyper-parameter controlling their relative importance (Section 2.3.3). It is finally applied to two WSL tasks,

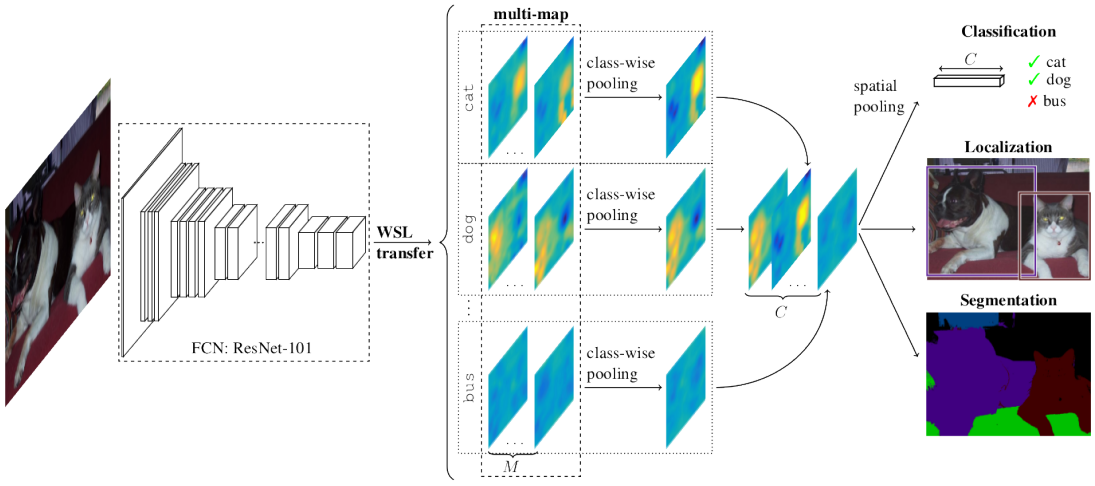



Figure 2.2 – **WILDCAT architecture.** It is integrated within a FCN backbone architecture to preserve spatial information throughout the network, leading to efficient region-level computation. Image features are encoded into multiple class modalities with a multi-map transfer layer, then globally pooled to yield image-wise values for WSL. WILDCAT can be applied to various WSL spatial tasks, such as object localization and semantic segmentation.

pointwise object localization and semantic segmentation (Section 2.3.4). We now delve into each point successively.

### 2.3.1 Fully Convolutional Architecture

The selection of relevant information within feature maps is a major issue in WSL. It impacts the localization of the learned representation and the precision of the results (*e.g.* semantic segmentation or object detection). We thus expect the resolution of the feature maps to be a key component for WILDCAT: finer maps keep more spatial resolution and lead to more specific regions (*e.g.* objects, parts).

To this end we exploit the recently introduced FCN ResNet-101 (He, X. Zhang, et al. 2016a) (left of Figure 2.2) that naturally preserves spatial information throughout the network. It also computes local features from all the regions in a single forward pass, without resizing them. Besides, ResNet architectures are effective at image classification while being parameter- and time-efficient (He, X. Zhang, et al. 2016a). This kind of architecture has been exploited to speed up computation and to produce accurate spatial predictions in fully supervised setups, *e.g.* in object detection (Dai, Y. Li, et al. 2016) and semantic segmentation (Dai, He, et al. 2016; Y. Li et al. 2017).

We use the publicly released model pre-trained on ImageNet dataset (Russakovsky et al. 2015) and remove the last layers (global average pooling and

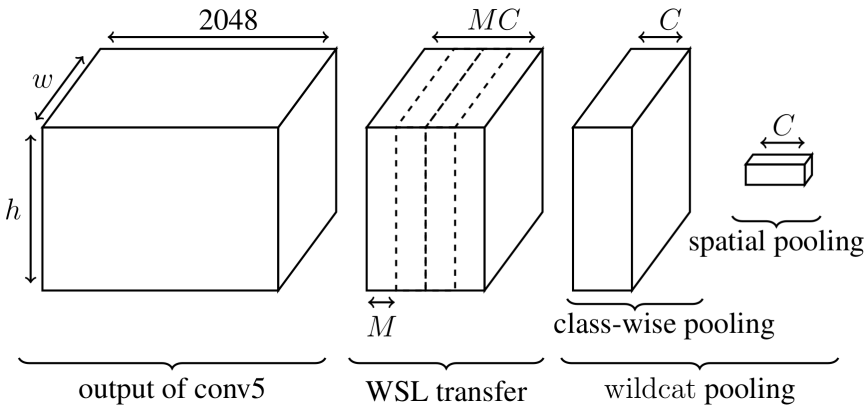



Figure 2.3 – **WILDCAT local feature encoding and pooling.** Class modalities are encoded with a multi-map WSL transfer layer and pooled separately for all classes. Local features are then aggregated with a global spatial pooling to yield a single score per class.

fully connected) to replace them with WSL transfer and wildcat pooling layers (Figure 2.3) described in the following.

### 2.3.2 Multi-Map Transfer Layer

We introduce a multi-map WSL transfer layer that learns multiple class-related modalities, encoded into $M$ feature maps per class through $1 \times 1$ convolutions (middle of Figure 2.2). The modalities are learned in a WSL fashion with only the image-level labels and the transfer layer keeps spatial resolution, key in WSL. We note $w \times h \times d$ the size of *conv5* maps of ResNet-101, which is $\frac{W}{32} \times \frac{H}{32} \times 2048$ for an original image of size $W \times H \times 3$ (He, X. Zhang, et al. 2016a). The transfer output is then of size $w \times h \times MC$ (Figure 2.3).

The $M$ modalities aim at specializing to different class-specific features, *e.g.* parts (Dai, Y. Li, et al. 2016; Dai, He, et al. 2016; Y. Li et al. 2017) (head and legs of dog in Figure 2.1) or views (Felzenszwalb et al. 2010; Girshick, Iandola, et al. 2015). We highlight differences with some specific encoding approaches: position-sensitive Region of Interest (RoI) pooling in Region-based Fully Convolutional Network (R-FCN) (Dai, Y. Li, et al. 2016) forces position-based specialization (relative to the object) while our method can also learn other kind of features, *e.g.* semantic parts (Figure 2.1). In the same way Deformable Part-based Model (DPM) (Felzenszwalb et al. 2010) learns only discriminating parts where our multi-map transfer model can find more general features, *e.g.* context. Furthermore, contrarily to the DPM where a different model is learned for each view, we share most of the computation within the FCN, which is more efficient. We note that when $M = 1$ this reduces to a standard classification layer, *i.e.* into $C$ classes.

### 2.3.3 WILDCAT Pooling

WILDCAT learns from image-level labels so we need a way to summarize all information contained in the feature maps for each class (right of Figure 2.2). We note that there are no more learned parameters in this pooling layers, which means we can directly interpret and visualize feature maps at this level (B. Zhou, Khosla, et al. 2016; Dai, Y. Li, et al. 2016).

We perform this in two steps (Figure 2.3): a class-wise pooling (Equation 2.1) that combines the $M$ maps from the multi-map transfer layer, then a spatial pooling module (Equation 2.2) that selects relevant regions within the maps to support predictions. This leads to WILDCAT pooling, a two-stage pooling operation to compute the score $s^c$ of class $c$:

$$\begin{cases} \bar{z}^c_{i,j} = \underset{m \in \{1,\dots,M\}}{\text{Cl. Pool}} \; z^{c,m}_{i,j} & (2.1) \\ s^c = \underset{(i,j) \in \{1,\dots,w\} \times \{1,\dots,h\}}{\text{Sp. Pool}} \; \bar{z}^c_{i,j} & (2.2) \end{cases}$$

where $z$ is the output of the transfer layer, Cl. Pool is the chosen class-wise pooling function and Sp. Pool is the spatial aggregation process.

**Class-wise pooling.** The first step consists in combining the $M$ maps for all classes independently, and is described in Equation 2.1 with a generic pooling function Cl. Pool. We use average pooling in the following. The maps are transformed from $w \times h \times MC$ to $w \times h \times C$ (Figure 2.3). When $M = 1$ this operation is not needed as each class is already represented by a single map.

We note that even if a multi-map followed by an average pooling is functionally equivalent to a single convolution (*i.e.* $M = 1$), the explicit structure it brings with $M$ modalities has important practical advantages making training easier. We empirically show that $M > 1$ yields better results than regular $M = 1$.

**Spatial pooling.** We now introduce our new spatial aggregation method implementing the second, spatial pooling step in Equation 2.2 for each map $c$:

$$s^c = \max_{\mathbf{h} \in \mathcal{H}_{k^+}} \frac{1}{k^+} \sum_{i,j} h_{i,j} \bar{z}^c_{i,j} + \alpha \left( \min_{\mathbf{h} \in \mathcal{H}_{k^-}} \frac{1}{k^-} \sum_{i,j} h_{i,j} \bar{z}^c_{i,j} \right) \tag{2.3}$$

where $\mathcal{H}_k$ is such that $\mathbf{h} \in \mathcal{H}_k$ satisfies $h_{i,j} \in \{0, 1\}$ and $\sum_{i,j} h_{i,j} = k$. It consists in selecting for each class $c$ the $k^+$ (resp. $k^-$) regions with the highest (resp. lowest) activations from input $\bar{z}^c$. The output $s^c$ for class $c$ of this layer is the weighted average of scores of all the selected regions. We only consider regions defined by single neurons in the convolutional feature maps.

Several similar MIL approaches have been used but our proposed model generalizes them in numerous of ways. The corresponding parameters are described in

| Global pooling function | $k^+$ | $k^-$ | $\alpha$ |
|---|---|---|---|
| Maximum (Oquab et al. 2015) | 1 | 0 | 0 |
| Top instances (W. Li et al. 2015) | $k$ | 0 | 0 |
| Label Proportion (Felix Yu et al. 2013) | $\rho n$ | 0 | 0 |
| GAP (B. Zhou, Khosla, et al. 2016) | $n$ | 0 | 0 |
| MANTRA (Durand, Thome, et al. 2015) | 1 | 1 | 1 |
| WELDON (Durand, Thome, et al. 2016) | $k$ | $k$ | 1 |

Table 2.1 – **Generalization of WILDCAT spatial pooling to other existing MIL approaches and variants with corresponding parameters.** $n$ is the total number of regions, $\rho$ is the proportion of positive labels in LLP, $k$ is an arbitrary number of regions to choose.

Table 2.1. The standard max-pooling MIL approach (Oquab et al. 2015) is obtained with only one element, and both top instance model (W. Li et al. 2015), Learning with Label Proportion (Felix Yu et al. 2013) and GAP (B. Zhou, Khosla, et al. 2016) can be obtained with more. Drawing from negative evidence (Parizi et al. 2015; Durand, Thome, et al. 2015; Durand, Thome, et al. 2016) we can incorporate minimum scoring regions to support classification and our spatial pooling function can reduce to the kMax+kMin layer of WELDON (Durand, Thome, et al. 2016).

Maximum and minimum scoring regions both are important for good results (Durand, Thome, et al. 2015; Durand, Thome, et al. 2016), but do not bring the same kind of information. We explore relative weighting of both types of regions by introducing a factor $\alpha$ which trades off relative importance between both terms. We hypothesize that maximum scoring regions are more useful for classification as they directly support the decision, while minimum scoring regions essentially act as regularization. With $\alpha < 1$, WILDCAT should focus more on discriminating regions and then better localize features than with $\alpha = 1$.

## Discussion

WILDCAT architecture is composed of a transfer layer followed by pooling. Since there are no parameters to learn in the pooling module, the transfer layer performs classification and it is easy to visualize heatmaps with direct localization of discriminating regions. We note that this kind of architecture is sometimes reversed (B. Zhou, Khosla, et al. 2016), and pooling is performed before the last fully connected layer, as in the original ResNet architecture (He, X. Zhang, et al. 2016a) for example. However this order requires an unnatural way of visualizing class-specific heatmaps (B. Zhou, Khosla, et al. 2016).

It is shown that if the spatial aggregation method is linear, *e.g.* GAP, then the order of both layers is not important (B. Zhou, Khosla, et al. 2016), but the two

configurations can behave differently with a non linear pooling function such as WILDCAT spatial pooling. The difference is more significant when $k^+ + k^-$ is low, *i.e.* when WILDCAT spatial pooling really differs from global average pooling. We evaluate the impact of this design choice and of the chosen pooling function in the experiments and show that our architecture yields better results.

### 2.3.4 WILDCAT Applications

**Training phase.** Our WILDCAT model is based on the backbone architecture ResNet-101 (He, X. Zhang, et al. 2016a). We initialize it from a model pre-trained on ImageNet (Russakovsky et al. 2015) and train it with Stochastic Gradient Descent (SGD) with momentum with image-level labels only. All the layers of the network are fine tuned. The input images are warped to a square size at a given scale. We use a multi-scale setup where a different model is learned for each scale and they are combined with Object Bank (L.-J. Li et al. 2010) strategy.

WILDCAT is designed to learn from image-level supervision only: the same training procedure is used for image classification, weakly supervised pointwise object detection and weakly supervised semantic segmentation. When learning WILDCAT, the gradients are backpropagated through the WILDCAT layer only within the $k^+ + k^-$ selected regions, all other gradients being discarded (Durand, Thome, et al. 2016). The selection of right regions for backpropagation is key to learn precisely localized features without any spatial supervision (Sun et al. 2016).

**Inference phase.** Predictions differ according to the task at hand. For image classification, prediction simply takes the single-value output of the network (like in training). Object detection and semantic segmentation require spatial predictions so we extract the class-specific maps before spatial pooling to keep spatial resolution. They are at resolution $\frac{1}{32}$ with respect to the input image for ResNet-101 architecture (He, X. Zhang, et al. 2016a). For weakly supervised pointwise object detection, we extract the region (*i.e.* neuron in the feature map) with maximum score for each class and use it for point-wise localization (Oquab et al. 2015; Bency et al. 2016). For weakly supervised semantic segmentation we compute the final segmentation mask either by taking the class with maximum score at each spatial position independently or by applying a Conditional Random Field (CRF) for spatial prediction as is common practice (Chen et al. 2015; Pathak, Krahenbuhl, et al. 2015).

## 2.4 Classification Experiments

To show the robustness of WILDCAT in very different image classification contexts, we evaluate it on six datasets, whose statistics are summarized in Table 2.2:

- PASCAL VOC 2007 and 2012 (Everingham et al. 2015) are standard datasets for object recognition. Each may contain multiple objects at any location and scale. Evaluation is done with mean Average Precision (mAP) as standard protocol;
- MIT67 (Quattoni et al. 2009) and 15 Scene (Lazebnik et al. 2006) are used for scene categorization, *i.e.* each image is associated with a single label for the full scene. Since there is only one prediction for each image, accuracy is used to score results;
- PASCAL VOC 2012 Action (Everingham et al. 2015) and MS COCO (T.-Y. Lin, Maire, et al. 2014) are two datasets for visual recognition where the context plays an important role. In action recognition, multiple labels may be applied to the same person, *e.g.* simultaneously walking and phoning. MS COCO is similar to PASCAL VOC datasets, but contains much more objects, with more variations in scale. In particular, there are lots of rather small objects, which are harder to detect. As for PASCAL VOC datasets, mAP is used for evaluation.

| Dataset | Train examples | Test examples | Classes | Metric |
|---|---|---|---|---|
| VOC07 | 5,011 | 4,952 | 20 | mAP |
| VOC12 | 11,540 | 10,991 | 20 | mAP |
| MIT67 | 5,360 | 1,340 | 67 | Accuracy |
| 15 Scene | 1,500 | 2,985 | 15 | Accuracy |
| VOC12Ac | 2,296 | 2,292 | 10 | mAP |
| MS COCO | 82,783 | 40,504 | 80 | mAP |

Table 2.2 – **Description of datasets**: number of train and test images, number of classes and evaluation metrics.

We first compare our model to state-of-the-art methods, then we analyze our contributions.

### 2.4.1 Comparison with State of the Art

We compare WILDCAT with several state-of-the-art object classification models (Table 2.3). The parameters of our model are fixed at $M = 4$ and $\alpha = 0.7$. We can point out a large improvement compared to deep features computed on the whole image with ResNet-101 (He, X. Zhang, et al. 2016a): 5.2 points on PASCAL VOC 2007 and 4.2 points on PASCAL VOC 2012. Note that these differences directly measure the relevance of the proposed WSL method, because WILDCAT is based on ResNet-101. We also compare our model to region selection approaches: DeepMIL

(Oquab et al. 2015), WELDON (Durand, Thome, et al. 2016) and RRSVM (Wei et al. 2016). Although using multiple regions as in (Oquab et al. 2015; Durand, Thome, et al. 2016; Wei et al. 2016) is important, we show here that we can further significantly improve performances by learning multiple modalities per category.

| Method | VOC07 | VOC12 |
|---|---|---|
| VGG16 (Simonyan et al. 2015) | 89.3 | 89.0 |
| DeepMIL (Oquab et al. 2015) | - | 86.3 |
| WELDON (Durand, Thome, et al. 2016) | 90.2 | - |
| ResNet-101 (He, X. Zhang, et al. 2016a) | 89.8 | 89.2 |
| ProNet (Sun et al. 2016) | - | 89.3 |
| RRSVM (Wei et al. 2016) | 92.9 | - |
| SPLeaP (Kulkarni et al. 2016) | 88.0 | - |
| WILDCAT (ours) | **95.0** | **93.4** |

Table 2.3 – **Classification results on object recognition datasets** (PASCAL VOC 2007 and 2012) in mAP (%). We used VOC evaluation server to evaluate on PASCAL VOC 2012.

In Table 2.4, we compare WILDCAT results for scene categorization with global image representations used for image classification: deep features (B. Zhou, Lapedriza, et al. 2014; He, X. Zhang, et al. 2016a), and global image representation with deep features computed on image regions: MOP CNN (Gong et al. 2014) and Compact Bilinear Pooling (Gao et al. 2016). It shows that seeking discriminative part regions is important, compared to incorporating background and non-informative parts into image representation. We also compare WILDCAT to existing part-based models including negative evidence during training (Parizi et al. 2015) and non-linear part classifiers combined with part-dependent soft pooling (Kulkarni et al. 2016). WILDCAT also outperforms WSL models with different spatial pooling strategies: 17.4 points with respect to GAP GoogLeNet (B. Zhou, Khosla, et al. 2016) which uses a GAP and 6.0 points with respect to WELDON (Durand, Thome, et al. 2016) which uses a kMax+kMin pooling on MIT67 dataset.

Finally, we report the performances of WILDCAT on context datasets in Table 2.5. We compare our model to ResNet-101 deep features (He, X. Zhang, et al. 2016a) computed on the whole image and WSL models for image classification: DeepMIL (Oquab et al. 2015), WELDON (Durand, Thome, et al. 2016) and ProNet (Sun et al. 2016). WILDCAT outperforms ResNet-101 by 8.1 and 8.2 points on both datasets, again validating our WSL model in this context.

| Method | 15 Scene | MIT67 |
|---|---|---|
| CaffeNet Places (B. Zhou, Lapedriza, et al. 2014) | 90.2 | 68.2 |
| MOP CNN (Gong et al. 2014) | - | 68.9 |
| Negative parts (Parizi et al. 2015) | - | 77.1 |
| GAP GoogLeNet (B. Zhou, Khosla, et al. 2016) | 88.3 | 66.6 |
| WELDON (Durand, Thome, et al. 2016) | 94.3 | 78.0 |
| Compact Bilinear Pooling (Gao et al. 2016) | - | 76.2 |
| ResNet-101 (He, X. Zhang, et al. 2016a) | 91.9 | 78.0 |
| SPLeaP (Kulkarni et al. 2016) | - | 73.5 |
| WILDCAT (ours) | **94.4** | **84.0** |

Table 2.4 – **Classification results on scene datasets** (15 Scene and MIT67) in multi-class accuracy (%).

| Method | VOC12Ac | MS COCO |
|---|---|---|
| DeepMIL (Oquab et al. 2015) | - | 62.8 |
| WELDON (Durand, Thome, et al. 2016) | 75.0 | 68.8 |
| ResNet-101 (He, X. Zhang, et al. 2016a) | 77.9 | 72.5 |
| ProNet (Sun et al. 2016) | - | 70.9 |
| WILDCAT (ours) | **86.0** | **80.7** |

Table 2.5 – **Classification results on context datasets** (PASCAL VOC 2012 Action and MS COCO) in mAP (%).

### 2.4.2 Further Analysis

We detail the impact of our contributions on three datasets: PASCAL VOC 2007, PASCAL VOC 2012 Action and MIT67. We present results for an input image of size $448 \times 448$px and $k^+ = k^- = 1$, but similar behaviors are observed for other scales and larger $k^+$ and $k^-$. By default, our model parameters $\alpha$ and $M$ are fixed to 1.

**Order of layers.** Firstly, to validate the design choice of the proposed WILDCAT architecture, we evaluate two different configurations (see discussion in Section 2.3.4):

(a) conv5 + conv + pooling (ours);

(b) conv5 + pooling + conv (B. Zhou, Khosla, et al. 2016).

These two configurations are different for the non-linear WILDCAT pooling scheme described in Section 2.3.3, and their comparison is reported in Table 2.6. We can see that our architecture (a) leads to a consistent improvement over architecture

(b) used in GAP (B. Zhou, Khosla, et al. 2016) on all three datasets, *e.g.* 1.7 points on PASCAL VOC 2007.

| Method | VOC07 | VOC12Ac | MIT67 |
|---|---|---|---|
| Architecture (a) | **89.0** | **78.9** | **69.6** |
| Architecture (b) | 87.3 | 77.5 | 68.1 |

Table 2.6 – **Analysis of pooling position.** Classification results for architectures (a) and (b).

Note that the two strategies of architecture have different interpretations: (a) classifies each region independently and then pools the region scores, whereas (b) pools the output of the convolution maps and then performs image classification on the pooled space.

**Impact of parameter $\alpha$.** We investigate the effect of the parameter $\alpha$ on classification performance. From the results in Figure 2.4, it is clear that incorporating negative evidence, *i.e.* $\alpha > 0$, is beneficial for classification, compared to standard max pooling, *i.e.* $\alpha = 0$. We further note that using different weights for maximum and minimum scores, *i.e.* $\alpha \neq 1$, yields better results than with $\alpha = 1$ from WELDON (Durand, Thome, et al. 2016), with best improvement of 1.6 points (resp. 2.0 and 1.8) with $\alpha = 0.6$ (resp. 0.7 and 0.8) on PASCAL VOC 2007 (resp. PASCAL VOC 2012 Action and MIT67). This confirms the relevance of using a relative weighting for negative evidence. Moreover our model is robust with respect to the value of $\alpha$.

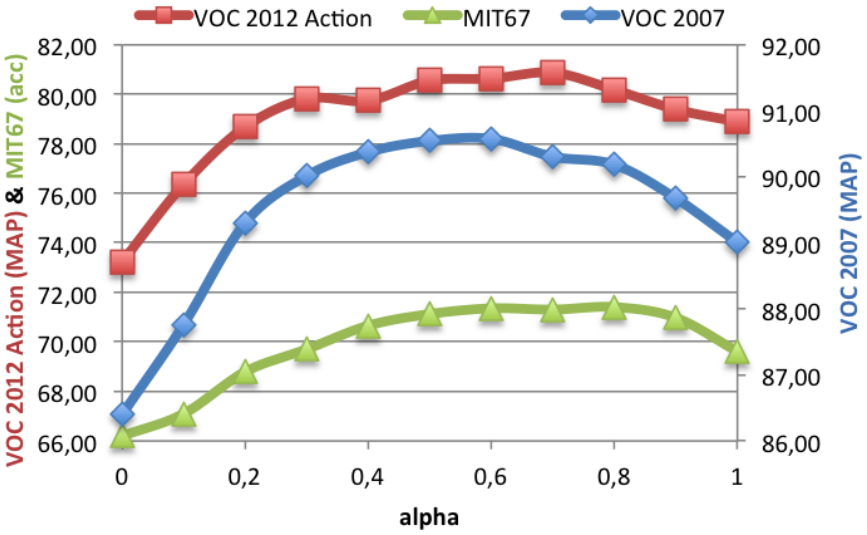



Figure 2.4 – **Analysis of parameter $\alpha$** on PASCAL VOC 2012 Action, MIT67 and PASCAL VOC 2007.

**Number of modalities.** Another important hyper-parameter of our model is the number of modalities ($M$) used in the multi-map transfer layer. The performances for different values of $M$ are reported in Table 2.7. Explicitly learning multiple modalities, *i.e.* $M > 1$, yields large gains with respect to a standard classification layer, *i.e.* $M = 1$ (Durand, Thome, et al. 2016). However encoding more modalities than necessary (*e.g.* $M = 16$) might lead to overfitting since the performances decrease. The best improvement is 3.5 points (resp. 4.3 and 3.5) with $M = 8$ (resp. 8 and 12) on PASCAL VOC 2007 (resp. PASCAL VOC 2012 Action and MIT 67). Examples of heatmaps for the same category are shown in Figure 2.5.

| | $M$ | | | | | |
|---|---|---|---|---|---|---|
| Dataset | 1 | 2 | 4 | 8 | 12 | 16 |
| VOC 2007 | 89.0 | 91.0 | 91.6 | **92.5** | 92.3 | 92.0 |
| VOCAction | 78.9 | 81.5 | 82.1 | **83.2** | 83.0 | 82.7 |
| MIT67 | 69.6 | 71.8 | 72.0 | 72.8 | **73.1** | 72.9 |

Table 2.7 – **Analysis of multi-map transfer layer** with respect to $M$ on PASCAL VOC 2007, PASCAL VOC 2012 Action and MIT67.

**Ablation study.** We perform an ablation study to illustrate the effect of each contribution. Our baseline is a WSL transfer with $M = 1$ and the spatial pooling with $\alpha = 1$. The results are reported in Table 2.8. From this ablation study, we see that both $\alpha = 0.7$ and $M = 4$ improvements separately result in large performance gains on all datasets, and that combining them further boosts performance: 0.4 points on PASCAL VOC 2007, 0.8 points on PASCAL VOC 2012 Action and 0.8 points on MIT67. This shows the complementarity of both these contributions.

| Model | | | Dataset | | |
|---|---|---|---|---|---|
| max + min | $\alpha = 0.7$ | $M = 4$ | VOC07 | VOC12Ac | MIT67 |
| ✓ | | | 89.0 | 78.9 | 69.6 |
| ✓ | ✓ | | 90.3 | 80.9 | 71.3 |
| ✓ | | ✓ | 91.6 | 82.1 | 72.0 |
| ✓ | ✓ | ✓ | **92.0** | **82.9** | **72.8** |

Table 2.8 – **Ablation study of WILDCAT** on PASCAL VOC 2007, PASCAL VOC 2012 Action (VOCAc) and MIT67. The results are different from results of Section 2.4.1 because only one scale is used for this analysis.

## 2.5 Weakly Supervised Experiments

In this section, we show that our model can be applied to various tasks, while being trained from global image labels only. We evaluate WILDCAT for two challenging weakly supervised applications: pointwise localization and segmentation.

### 2.5.1 Weakly Supervised Pointwise Object Localization

We evaluate the localization performances of our model on PASCAL VOC 2012 validation set (Everingham et al. 2015) and MS COCO validation set (T.-Y. Lin, Maire, et al. 2014). The performances are evaluated with a point-based object localization metric: if a prediction for a class falls inside the ground truth bounding box of an object of this class, it is counted as correct (Oquab et al. 2015). This metric measures the quality of the detection, while being less sensitive to misalignments compared to other metrics such as Intersection over Union (IoU) (Everingham et al. 2015), which requires the use of additional steps (*e.g.* bounding box regression).

WILDCAT localization performances are reported in Table 2.9. We can notice an important improvement between WILDCAT and MIL-based architecture DeepMIL (Oquab et al. 2015), which confirms the relevance of our spatial pooling function. In spite of its simple and multipurpose architecture, our model outperforms by a large margin the complex cascaded architecture of ProNet (Sun et al. 2016). It also outperforms the recent weakly supervised model (Bency et al. 2016) by 3.2 points (resp. 4.2) on PASCAL VOC 2012 (resp. MS COCO), which use a more complex strategy than our model, based on search-trees to predict locations.

| Method | VOC12 | MS COCO |
|---|---|---|
| DeepMIL (Oquab et al. 2015) | 74.5 | 41.2 |
| ProNet (Sun et al. 2016) | 77.7 | 46.4 |
| WSLocalization (Bency et al. 2016) | 79.7 | 49.2 |
| WILDCAT (ours) | **82.9** | **53.4** |

Table 2.9 – **Pointwise object localization results** on PASCAL VOC 2012 and MS COCO in mAP (%).

### 2.5.2 Weakly Supervised Semantic Segmentation

We evaluate our model on the PASCAL VOC 2012 image segmentation dataset (Everingham et al. 2015), consisting of 20 foreground object classes and one background class. We train our model with the train set (1,464 images) and the extra

annotations provided by (Hariharan et al. 2011) (resulting in an augmented set of 10,582 images), and test it on the validation set (1,449 images). The performance is measured in terms of pixel mean Intersection over Union (mIoU) (pixel IoU averaged across the 21 categories). As in existing methods, we add a fully connected CRF (Krähenbühl et al. 2011) to post-process the final output labeling.

The result of our method is presented in Table 2.10. We compare it to weakly supervised methods that only use image labels during training. We can see that WILDCAT without CRF outperforms existing weakly supervised models. We note a large gain with respect to MIL models based on (soft-)max pooling (Pathak, Shelhamer, et al. 2015; Pinheiro et al. 2015), which validates the relevance of our pooling for segmentation. The improvement between WILDCAT with CRF and the previous best model is 7.1 points. This confirms the ability of our model to learn discriminative and accurately localized features. We can note that all the methods evaluated in Table 2.10 have comparable complexity.

| Method | mIoU |
|---|---|
| MIL-FCN (Pathak, Shelhamer, et al. 2015) | 24.9 |
| MIL-Base+ILP+SP-sppxl (Pinheiro et al. 2015) | 36.6 |
| EM-Adapt + CRF (Papandreou et al. 2015) | 33.8 |
| CCNN + CRF (Pathak, Krahenbuhl, et al. 2015) | 35.3 |
| WILDCAT (ours) | 39.2 |
| WILDCAT + CRF (ours) | **43.7** |

Table 2.10 – **Semantic segmentation results** on PASCAL VOC 2012 in mIoU (%).

With a quite more complex strategy, contemporaneous SEC model (Kolesnikov et al. 2016) presents impressive results (50.7 mIoU). The training scheme incorporates different terms, which are specifically tailored to segmentation: one enforces the segmentation mask to match low-level image boundaries, another one incorporates prior knowledge to support predicted classes to occupy a certain image proportion. In contrast, WILDCAT uses a single model which is trained in the same manner for the three tasks, *i.e.* classification, localization and segmentation.

### 2.5.3 Visualization of Results

In Figure 2.5, we show predicted segmentation masks for four images. Compared to ground truth (column (b)), we can see that our predicted segmentation masks (column (c)) are always relevant, except for the last example where the rails and the train are glued together. The heatmaps from the same class (columns (d) and (e)) show different modalities learned by our model. When successful, they focus on different parts of the objects. For example, on the first row, the heatmap

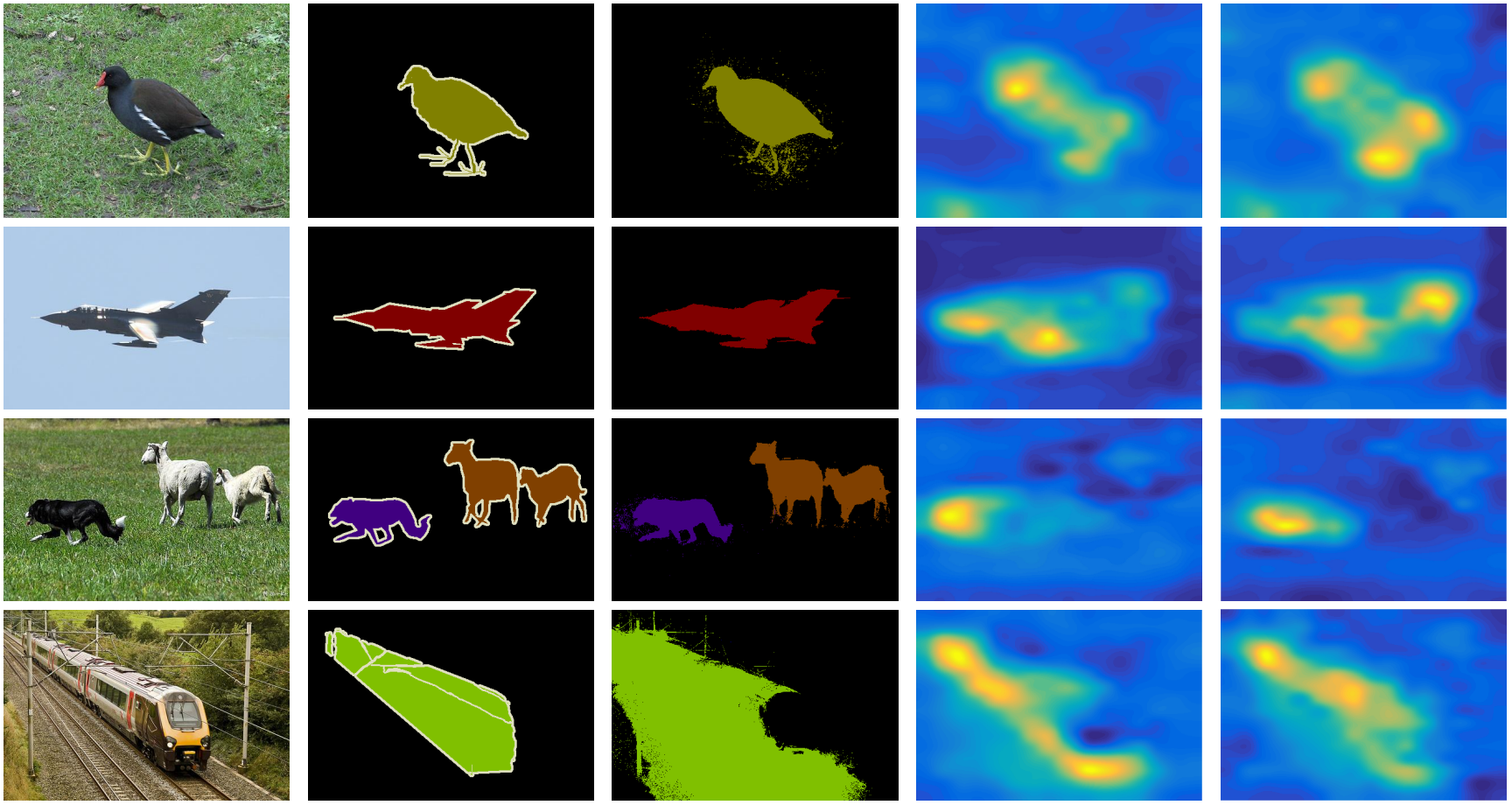
(d) focuses on the head of the bird whereas the heatmap (e) focuses on the legs and the tail.

## 2.6 Conclusion

In this chapter, we have introduced WILDCAT, a deep MIL model for WSL from image-level supervision. It learns localized representations thanks to two components: a global pooling function extending previous negative evidence models (Durand, Thome, et al. 2016) by separating contributions from positive and negative regions, and a multi-map transfer layer building structure related to class modalities into the representation. By integrating WILDCAT within a FCN backbone architecture (He, X. Zhang, et al. 2016a), it can efficiently be end-to-end trained for spatial tasks such as object localization and semantic segmentation in a straightforward way, although using global labels only.

We have shown that learning multiple ($M$) feature maps related to class modalities, as well as selecting several positive ($k^+$) and negative ($k^-$) regions is beneficial for WSL. In all experiments, these hyper-parameters were set to default values that were found optimal in average across all classes. However, different values should intuitively be attributed to different classes. Indeed, some classes are more complex than others, and would require more modalities or more regions to be fully described, while it would not be desired for simpler classes. As future work, it then seems interesting to learn these values separately for all classes during training.

In the next chapter, we consider more supervision and exploit the WSL context for learning at the object level, with bounding box annotations. The goal is to learn part-based representations that can better adapt to shapes of objects, in a latent way, *i.e.* without part annotation.



(a) image (b) ground truth (c) WILDCAT prediction (d) heatmap 1 (e) heatmap 2

Figure 2.5 – **Segmentation examples on PASCAL VOC 2012.** Our prediction is correct except for the train (last row) where our model aggregated rails and train regions. For objects as *bird* or *plane*, one can see how two heatmaps (heatmap 1 (d) and heatmap 2 (e) representing the same class: respectively *bird, aeroplane, dog* and *train*) succeed to focus on different but relevant parts of the objects.

CHAPTER 3

# LEARNING PART-BASED REPRESENTATIONS FROM OBJECT-LEVEL SUPERVISION

## Contents




***Chapter abstract***

*This chapter relates to standard object detection task, which is commonly decomposed into object recognition and bounding box regression sub-tasks. We are interested in learning deformable part-based representations,* i.e. *composed of multiple local descriptors that are dynamically positioned on objects during inference. This kind of representation is more flexible than traditional bounding box-based ones to describe objects, and is especially adapted to non-rigid ones. We propose DP-FCN, a deep object detector exploiting this idea to learn fine representations. It aligns parts to discriminative elements of objects in a principled latent way,* i.e. *without part annotation, with an extension of Multiple-Instance Learning (MIL) framework. This is beneficial for both sub-tasks of detection. By aligning parts, the recognition step is more robust to local transformations, since the representation changes accordingly and is more invariant. This also gives access to the positions of parts relative to each*

*other, which is a rich geometric information about shapes of objects. This is then leveraged to improve the accuracy of localization, by fitting bounding boxes more tightly around objects. Structure is further introduced in the model by taking interactions between all parts into account. This leads to a joint structured optimization of all parts, improving their positioning and the representations of objects. Localization is also refined in the same way, by exploiting correlations in part positions.*

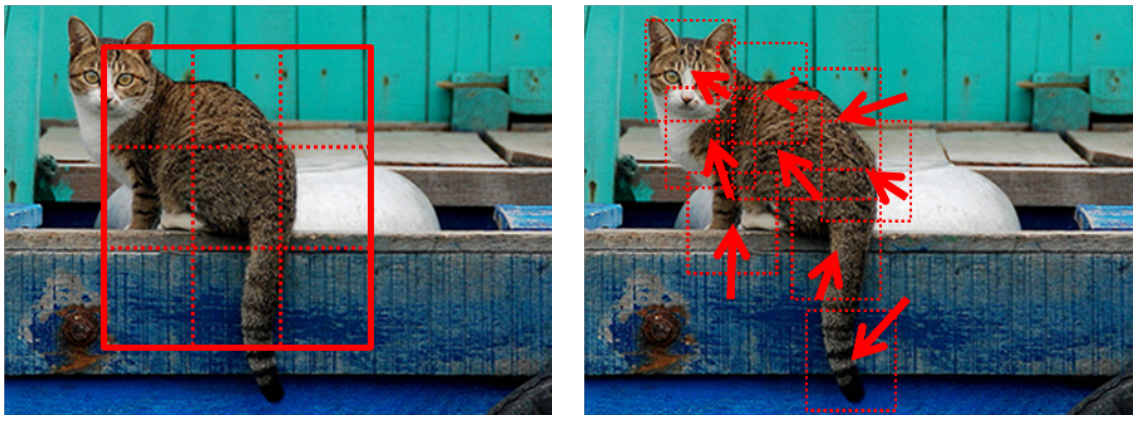
*The work in this chapter has led to the publication of a conference paper and a journal paper:*

- Taylor Mordan, Nicolas Thome, Matthieu Cord, and Gilles Henaff (2017). "Deformable Part-based Fully Convolutional Network for Object Detection". In: *Proceedings of the British Machine Vision Conference (BMVC)*, Best Science Paper Award;
- Taylor Mordan, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018a). "End-to-End Learning of Latent Deformable Part-Based Representations for Object Detection". In: *International Journal of Computer Vision (IJCV)*.

## 3.1 Introduction

Recent years have witnessed a great success of Deep Learning (DL) with deep Convolutional Neural Networks (ConvNets) (LeCun et al. 1989; Krizhevsky et al. 2012) in several visual tasks, and object detection is no exception, as summarized in Section 1.2.3. It is a major task of Computer Vision (CV), mixing semantic recognition and spatial localization, and one of the first step to scene understanding. We focus on deep region-based object detectors (Girshick, Donahue, et al. 2014) as they form the approach with best results, see Section 1.2.3 for a more detailed presentation. The idea is to use region proposals, either from an external algorithm (Girshick 2015) or learned by the network (S. Ren et al. 2015), to focus computation on promising areas. Networks are applied to whole images, and carry region-level computations in an efficient way through a Region of Interest (RoI) pooling layer selecting features around these region proposals. In this chapter, we adapt the Weakly Supervised Learning (WSL) context from Chapter 2 to region-based object detection. This time, supervision is given at the object level with bounding box annotations, and finer representation is learned within boxes as latent information.

Although they yield excellent results in object detection, region-based deep ConvNets still present a few issues that need to be addressed. As already noted in Section 1.2.3, networks are usually initialized with models pre-trained for image classification on ImageNet dataset (Russakovsky et al. 2015), and are therefore prone to suffer from mismatches between classification and detection tasks. As



(a) original region (b) deformed region

Figure 3.1 – **Illustration of deformations.** Regions are divided into regular grids (a) and all cells are moved from their initial positions to adapt to the shape of the object and better describe it (b), improving both recognition and localization.

an example, pooling layers bring invariance to local transformations and help learning more robust features for classification, but they also reduce the spatial resolution of feature maps and make the network less sensitive to the positions of objects within regions (Dai, Y. Li, et al. 2016), both of which are bad for accurate localization. Furthermore, the use of bounding boxes with fixed rectangular shape limits the representation of objects, especially when objects are not rigid and parts can move relative to each other.

A possible solution is to use part-based representations, as popularized by the seminal work of Deformable Part-based Model (DPM) (Felzenszwalb et al. 2010), presented in Section 1.2.4. It comprises multiple local templates that are dynamically localized with a Latent Support Vector Machine (LSVM) to detect parts of objects. This kind of representation encodes finer information about objects and is more flexible than global bounding box ones, leading to better fits of objects. It is especially useful with deformable objects since the representation adapts accordingly. It is noticeable that DPM is a historical model that had a huge success in object detection before the revolution of DL (Everingham et al. 2015). DPM recently received the PAMI Longuet-Higgins Prize at CVPR 2018 for fundamental contribution in CV, and its idea are still currently under research within DL.

This chapter introduces a new strategy to learn deformable part-based representations in ConvNets for object detection, as illustrated in Figure 3.1. Parts are optimized with a deformable part-based RoI pooling function, based on a Multiple-Instance Learning (MIL) framework applied at the object level, *i.e.* with bounding boxes but without part annotations. Aligning parts builds invariance to local transformations of objects, therefore improving recognition, but also gives

access to their configurations (*i.e.* their positions relative to each other, which can be observed in Figure 3.1 (b)), which brings important geometric information about objects, *e.g.* their shapes. We also leverage this in a dedicated localization module to better fit objects. These two modules are integrated into a deep region-based object detector architecture, which we name Deformable Part-based Fully Convolutional Network (DP-FCN). It generalizes previous region-based models by integrating latent deformable part-based representations of objects. These are inspired by DPM, where all parts are conditionally independent from each other. We further improve DP-FCN by removing this assumption and introducing structure on deformations. Parts are not considered separately anymore, but as jointly forming whole objects. This is done in both modules, by explicitly taking relative interactions of parts into account.

## 3.2 Related Work

Several works have investigated the use of part-based representations with ConvNets. Parts are often used for fine-grained recognition with different methods: learning a module for localizing and aligning parts with respect to templates before classifying them (D. Lin et al. 2015), finding part proposals from activation maps and learning a graphical model to recognize objects (Simon et al. 2015), using two sub-networks for detection and classification of parts (H. Zhang et al. 2016), considering parts as a vocabulary of latent discriminative features decoupled from the task and learning them in an unsupervised way (Sicre et al. 2017). Usage of parts is also common in semantic segmentation (P. Wang et al. 2015a; Dai, He, et al. 2016; Y. Li et al. 2017).

Parts are introduced for detection by learning part models and combining them with geometric constraints for scoring (N. Zhang, Donahue, et al. 2014). They are learned in a strongly supervised way, *i.e.* with part annotations. Although manually defining parts can be more interpretable, it requires longer annotation time (see Figure 1.6) and is likely sub-optimal for detection as they might not correspond to most discriminative elements. For these reasons, we are interested in learning parts in a latent way, with an extension of MIL framework, as in the original DPM.

**Deformable Part-based Model.** The core idea behind DPM (Felzenszwalb et al. 2010) is to represent each class by a root filter describing whole appearances of objects and a set of part filters at finer scales to model local parts. Each part filter is assigned to an anchor point, defined relative from the root, and move around during detection to model deformations of objects and best fit them. A regularization is further introduced in the form of a deformation cost penalizing large displacements. Each part is then optimizing a trade-off between maximizing

detection score and minimizing deformation cost. Final detection output combines scores from root and all parts. Accurate localization is done with a post-processing step.

Several extensions have been proposed to DPM, *e.g.* using a second hierarchical level of parts to finely describe objects (L. Zhu et al. 2010), sharing part models between classes (Ott et al. 2011), learning from strongly supervised annotations (*i.e.* at the part level) to get a better model (Azizpour and Laptev 2012), exploiting segmentation clues to improve detection (Fidler et al. 2013).

DPM is in contrast with region-based deep ConvNets: while the latter relies on strong features learned directly from pixels and exploit region proposals to focus on interesting areas of images, DPM explicitly takes into account geometry of objects by optimizing a graph-based representation and is usually applied in a sliding window fashion over weak image features. Both approaches exploit different hypotheses and seem therefore complementary.

**Integration of Deformable Part-based Model with deep ConvNets.** Several attempts have been made to integrate DPM with deep ConvNets. The first ones (Savalle et al. 2014; Girshick, Iandola, et al. 2015; Wan et al. 2015) simply exploited deep features learned by an AlexNet (Krizhevsky et al. 2012) to use them with DPM. However, the fully connected layers are hard to apply in this context, since they heavily reduce spatial information needed for accurate part localization, and they cannot be used after part optimization. They are then removed completely, but this adversely affects performance. While the two approaches seem complementary, it is not straightforward to combine them efficiently. We propose to solve these issues by using a Fully Convolutional Network (FCN) architecture as discussed in the following.

**Fully Convolutional Networks.** Since they only contain convolutional layers, FCNs (Long et al. 2015) offer natural solutions to keep spatial resolution. This is an important property for all spatial tasks, which require feature maps to encode spatial information. Recently, FCNs have been successfully used to perform such tasks, *e.g.* object detection (Dai, Y. Li, et al. 2016), instance segmentation (Dai, He, et al. 2016; Y. Li et al. 2017), or WSL (Durand, Mordan, et al. 2017), in a efficient way, with almost all computation shared across image and only few per-region computation. In object detection, Region-based Fully Convolutional Network (R-FCN) (Dai, Y. Li, et al. 2016) achieves competitive results with this approach. Compared to previous Fast Region-based Convolutional Neural Network (R-CNN) (Girshick 2015), the sub-networks after RoI pooling are here reduced at minimum to have very light per-region computation. Classification and localization for each region are then achieved by encoding information into several feature maps, processed by a position-sensitive RoI pooling layer, rather than in the following corresponding sub-networks.

We improve the R-FCN architecture by generalizing the position-sensitive RoI pooling function with an extension of MIL, to optimize displacements of parts in a latent way. Since all convolutional layers are shared across images, it is no longer necessary to remove layers from a pre-trained network, and this solves the aforementioned issues of integrating DPM into deep ConvNets.

**Structured prediction.** Joint optimization of multiple variables is often performed to bring spatial coherence in tasks with structured predictions, such as semantic segmentation (Krähenbühl et al. 2011; Chen et al. 2015; Zheng et al. 2015). For this application, this yields improved results compared to independently classifying each pixel, by filtering out spatially isolated labels or taking more context into account. The optimization problem often being challenging, a solution is to cast it as an inference over a Conditional Random Field (CRF) tailored to the problem, for which there exist several algorithms.

For semantic segmentation, an efficient inference algorithm relying on Mean Field approximation has been proposed for fully connected CRFs (Krähenbühl et al. 2011). It shows improvements with joint optimization of all pixels with respect to independent prediction at each location, while keeping computational requirements low. The same algorithm has then been used in a number of following works in semantic segmentation (Chen et al. 2015; Zheng et al. 2015). In particular, it can be integrated as layers within networks so that models are learned in an end-to-end way with CRFs (Zheng et al. 2015). These can then influence training, as they are not relegated to post-processing anymore. This approach has been generalized by learning deep embeddings (Chandra et al. 2017), allowing exact inference over fully connected CRFs, and applied to other tasks than semantic segmentation, such as saliency estimation and human part segmentation.

With DP-FCN, we propose to cast the computation of deformations of regions as a CRF optimization, so that all parts are optimized jointly and their interactions are expressed in the model. Here, the optimization is performed at the core of the network during inference, and is not just a post-processing step integrated into training.

## 3.3 Deformable Part-based Fully Convolutional Networks

In this section, we present our model Deformable Part-based Fully Convolutional Network (DP-FCN), a deep region-based network for object detection. It represents regions with several parts it aligns by explicitly optimizing their positions in a deformable part-based RoI pooling function. This is based on a Multiple-Instance Learning (MIL) framework applied at the object level, *i.e.*

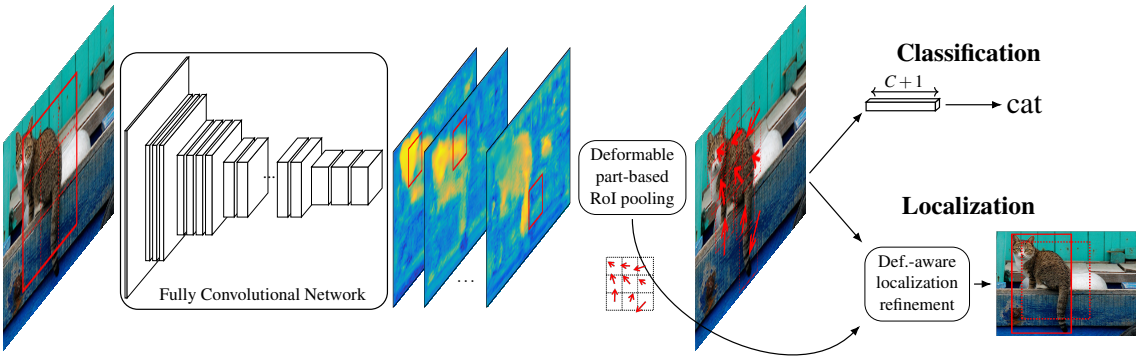



Figure 3.2 – **Architecture of DP-FCN.** It is composed of a FCN to extract dense feature maps with high spatial resolution, a deformable part-based RoI pooling layer to compute a representation aligning parts and two sibling classification and localization prediction branches. Initial rectangular region is deformed to focus on discriminative elements of object. Alignment of parts brings invariance for classification and geometric information refining localization *via* a deformation-aware localization module.

without part annotations, and generalizes previous RoI pooling functions with deformations. This alignment is learned end-to-end and improves both classification and localization. First, the part-based representations are more invariant to local transformations of objects, easing the classification. Once optimized, configurations of parts give important information about the geometry of objects, *e.g.* their shapes, which is leveraged in a deformation-aware localization module to better fit objects. We also introduced an improved version, named DP-FCN2.0, which additionally models interactions between all pairs of parts, benefiting both sub-tasks. This is done with an in-network Conditional Random Field (CRF) inference integrated within the RoI pooling function to add constraints into the optimization problem and better optimize part positions, and with a bilinear product for bounding box regression to model correlations between part positions. These ideas can be inserted into most of state-of-the-art network architectures to improve performances.

The complete architecture is depicted in Figure 3.2 and is composed of three main modules: (i) a Fully Convolutional Network (FCN) applied on whole images (Section 3.3.1), (ii) a deformable part-based RoI pooling layer (Section 3.3.2), and (iii) two sibling prediction layers for classification and localization (Section 3.3.3).

The work closest to ours is Deformable ConvNet (Dai, Qi, et al. 2017), a concurrent model which also exploits deformations to adapt to shapes of objects. While the ideas behind it are similar to ours, deformations are computed in a different way. Deformable ConvNet obtains deformations by using convolutional layers to

estimate them, whereas we cast it as an optimization problem and solve it. While the other approach is more general, in that it can be applied to convolutional layers in addition to RoI pooling layers, the solutions we propose here are more controllable and can be tuned to specific purposes.

We now describe all three parts of our model in more details.

### 3.3.1 Fully Convolutional Feature Extractor

Our model relies on a FCN (He, X. Zhang, et al. 2016a; Zagoruyko and Komodakis 2016; Xie et al. 2017) as backbone architecture, as this kind of network enjoys several practical advantages, leading to several successful models (Dai, Y. Li, et al. 2016; Y. Li et al. 2017; Durand, Mordan, et al. 2017). First, it allows to share most computation on whole images and to reduce per-RoI layers, as noted in R-FCN (Dai, Y. Li, et al. 2016). Second and most important to our work, it directly provides feature maps linked to the task at hand (*e.g.* detection heatmaps, as illustrated in the middle of Figure 3.2 and on the left of Figure 3.3) from which final predictions are simply pooled, as done by (Dai, Y. Li, et al. 2016; Durand, Mordan, et al. 2017). Within DP-FCN, inferring the positions of parts for a region is done with a particular kind of RoI pooling that we describe in Section 3.3.2.

The fully convolutional structure is therefore suitable for computing responses of all parts for all classes as a single map for each of them. A corresponding structure is used for localization. The complete representation for a whole image (classification and localization maps for each part of each class) is obtained with a single forward pass and is shared between all regions of the same image, which is very efficient.

Since relocalization of parts is done within feature maps, the resolution of these maps is of practical importance (Savalle et al. 2014; Girshick, Iandola, et al. 2015; Wan et al. 2015). FCNs contain only spatial layers and are therefore well suited for preserving spatial resolution, as opposed to networks ending with fully connected layers (Krizhevsky et al. 2012; Simonyan et al. 2015). Specifically, if the stride is too large, deformations of parts might be too coarse to describe objects correctly. We reduce it by using dilated convolutions (Chen et al. 2015; Long et al. 2015; Fisher Yu et al. 2016) on the last convolution block and skip pooling (Bell et al. 2016; Kong et al. 2016; Zagoruyko, Lerer, et al. 2016) to combine the last three.

### 3.3.2 Deformable Part-based RoI Pooling

The aim of this layer is to divide region proposals into several parts and to locally relocalize these to best match shapes of objects (as illustrated in Figure 3.1). Each part then models a discriminative local element and is to be aligned at the corresponding location within the image. This deformable part-based representation

is more invariant to transformations of objects because the parts are positioned accordingly and their local appearances are stable (Felzenszwalb et al. 2010). This is especially useful for non-rigid objects, where a box-based representation must be sub-optimal.

The separation of a region $R$ into parts is done with a regular grid of fixed size $I \times J$ fitted to it (Girshick 2015; Dai, Y. Li, et al. 2016). Each cell $(i, j)$ is then interpreted as a distinct part $R_{i,j}$. This strategy is simple yet effective (L. Zhu et al. 2010; Wan et al. 2015). Since the number of parts (*i.e.* $IJ$) is fixed as a hyper-parameter, it is easy to have a complete detection heatmap $z_{i,j,c}$ already computed for each part $(i, j)$ of each class $c$ (left of Figure 3.3). Part locations then only need to be optimized within corresponding maps.

The deformation of parts allows them to slightly move around their reference positions (partitions of the initial regions), selects the optimal latent displacements, and pools values from selected locations. During training, deformations are optimized without part-level annotations, *i.e.* only box-level annotations are needed, just as in the traditional object detection task. Displacements computed during the forward pass are stored and used to backpropagate gradients at the same locations. We further note that the deformations are computed for all parts and classes independently. However, no deformation is computed for the *background* class: they would not bring any relevant information as there is no discriminative elements for this class. The same displacements of parts are used to pool values from the localization maps.

We present two different strategies for computing deformations in the next sections. The first one considers each part independently from others. While this is highly efficient, it might miss complex relations between parts. In contrast, the second method performs a joint optimization on all parts simultaneously and takes interactions between parts into account by leveraging a CRF formulation. It is then able to model object geometries more finely.

#### 3.3.2.1 Independent Deformations of Parts

This first approach draws ideas from the original DPM (Felzenszwalb et al. 2010) and is applied separately to all parts. For a part $(i, j)$ of a region $R$ and a class $c$, the set $\Delta^R_{i,j}$ of possible displacements $\delta = (\delta x, \delta y)$ is such that the part $R_{i,j}$ still stays within the feature map $z$ after moving by $\delta$. We then define the score $S^R_{i,j,c}(\delta)$ of these part and class for a displacement $\delta \in \Delta^R_{i,j}$ as the value pooled at the new location ($R_{i,j}$ offset by $\delta$) and penalized by the magnitude of the displacement:

$$S^R_{i,j,c}(\delta) = \underset{(x,y)\in R_{i,j}}{\text{Pool}} \, z_{i,j,c}(x + \delta x, y + \delta y) - \lambda^{def} \|\delta\|_2^2 \tag{3.1}$$

where $\lambda^{def}$ represents the strength of the regularization (bias toward small deformations), and Pool is an average pooling as in (Dai, Y. Li, et al. 2016), but any

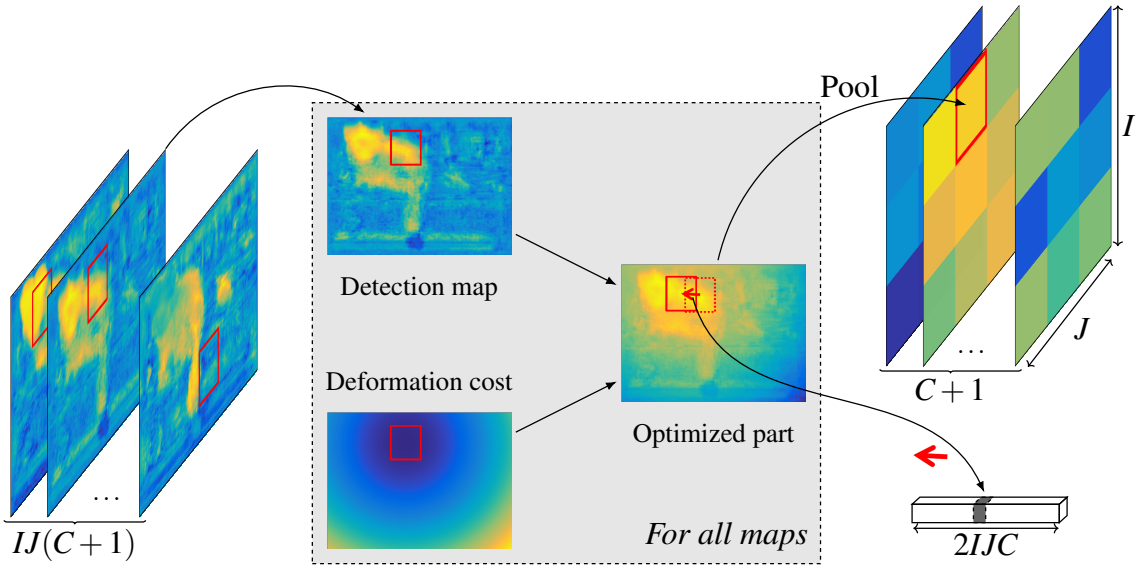



Figure 3.3 – **Deformable part-based RoI pooling with independent deformations.** Each input feature map corresponds to a part of a class (or *background*). Positions of parts are optimized separately within detection maps with deformation costs as regularization, and values are pooled within parts at the new locations. Output includes a map for each class and the computed displacements of parts, to be used for localization.

pooling function could be used instead. Here, the deformation cost is the squared distance of the displacement on the feature map, but other functions could be used equally. The deformable part-based RoI layer consists in maximizing this quantity with respect to the displacement, and therefore optimizes a trade-off between maximizing the score on the corresponding feature map and minimizing the displacement from the reference position (see Figure 3.3). Its output $p_c^R(i,j)$ then writes:

$$p_c^R(i,j) = \max_{\delta \in \Delta_{i,j}^R} \left[ S_{i,j,c}^R(\delta) \right] \tag{3.2}$$

$$= \max_{\delta \in \Delta_{i,j}^R} \left[ \underset{(x,y) \in R_{i,j}}{\text{Pool}} \; z_{i,j,c}(x+\delta x, y+\delta y) - \lambda^{def} \|\delta\|_2^2 \right]. \tag{3.3}$$

While Equation 3.3 is used to compute the output of the layer for part $(i,j)$ of region $R$ and class $c$, it also gives the displacement $d_c^R(i,j) = \left(dx_c^R(i,j), dy_c^R(i,j)\right)$ for that part: it is the argmax of Equation 3.3, *i.e.* the $\delta = (\delta x, \delta y)$ maximizing it. Those displacements are extracted from the layer to be used for localization thereafter (see Section 3.3.3). We emphasize that this formulation does not require any annotations about positions of parts, and can therefore be used in the standard object detection setup (*i.e.* with bounding boxes only).

$\lambda^{def}$ is directly linked to the magnitudes of the displacements of parts, and therefore to the deformations of RoIs too, by controlling the squared distance regularization (*i.e.* preference for small deformations). Increasing it puts a higher weight on regularization and effectively reduces displacements of parts, but setting it too high prevents parts from moving and removes the benefits of our approach. It is noticeable this deformable part-based RoI pooling is a generalization of the position-sensitive RoI pooling from (Dai, Y. Li, et al. 2016). Setting $\lambda^{def} = +\infty$ clamps all displacements $d_c^R(i,j)$ to $(0,0)$, leading to the formulation of position-sensitive RoI pooling:

$$p_c^R(i,j) = \underset{(x,y)\in R_{i,j}}{\text{Pool}} \; z_{i,j,c}(x,y). \tag{3.4}$$

On the other hand, setting $\lambda^{def} = 0$ removes regularization and parts are then free to move. With $\lambda^{def}$ too low, the results decrease, indicating that regularization is practically important. However, the results appeared to be stable within a large range of values of $\lambda^{def}$. Additionally, optimization of $\delta$ is performed by brute force in a limited range and not the whole image, *i.e.* the sets $\Delta_{i,j}^R$ are restricted to their intersections with a centered ball of small radius. With $\lambda^{def}$ not too small, the regularization effectively restricts displacements to lower values, leaving the results of pooling unchanged. In all experiments, we use $\lambda^{def} = 0.3$.

We further normalize the displacements $dx_c^R$ and $dy_c^R$ by the heights and widths of parts respectively to make the layer invariant to the scales of the images and regions. Indeed, the parts should move to the same positions relative to the objects, regardless of the scales at which they appear in the images and irrespective of any scaling factor applied to the images. We also normalize the classification feature maps before forwarding them to deformable part-based RoI pooling layer to ensure classification and regularization terms are comparable. We do this by $L_2$-normalizing at each spatial location the block of $C+1$ maps for each part separately, *i.e.* replacing $z$ from Equation 3.3 with

$$\bar{z}_{i,j,c}(x,y) = \frac{z_{i,j,c}(x,y)}{\sqrt{\sum_{c'} z_{i,j,c'}(x,y)^2}}. \tag{3.5}$$

#### 3.3.2.2 CRF-based Joint Deformations of Parts

The second strategy to compute deformations jointly considers all parts in a single optimization problem. All displacements are inferred simultaneously, so that it is possible to model dependencies between them and enforce consistency. We then have a fully connected graphical model, *i.e.* displacement of a given part is influenced by those of all other parts. This is in contrast with the independent deformations from Section 3.3.2.1 which uses a star model, *i.e.* parts are conditionally independent from each other given the whole region, like the original DPM (Felzenszwalb et al. 2010).

We do this by casting the optimization problem into a Conditional Random Field (CRF) inference over displacements of parts within regions. We define original unary and pairwise potentials by hand so that the CRFs act as a regularization and lead to a more robust part alignment stage. By integrating the CRF inference algorithm within the deformable part-based RoI pooling layer, *i.e.* the inference is carried out for all regions at each forward pass, we are still able to perform end-to-end training on Graphics Processing Unit (GPU) with a moderate overhead.

A different CRF is instantiated for each region $R$ and class $c$ (but for the *background* class as no deformations are computed), and they are all optimized in parallel during forward passes. There are $I \times J$ variables $D_c^R(i,j)$ considered here, each associated with a given part $(i,j)$ and indicating its displacement $d_c^R(i,j)$. The Gibbs probability distribution of the CRF conditioned on an image $I$ is then

$$P\left(D_c^R = d_c^R | I\right) = \frac{1}{Z_c^R(I)} \exp\left(-E_c^R(d_c^R|I)\right) \tag{3.6}$$

with $Z_c^R$ the partition function and $E_c^R$ the corresponding Gibbs energy (Lafferty et al. 2001). From now on, we drop the $R$ and $c$ notations as well as the conditioning on image $I$ for convenience.

We use the fully connected CRF formulation of (Krähenbühl et al. 2011) to model dependencies between all pairs of parts. The Gibbs energy $E$ for displacements $d$ then takes the form

$$E(d) = \sum_{i,j} \phi_u\left(d(i,j)\right) + \sum_{(i,j)<(i',j')} \phi_p\left(d(i,j), d(i',j')\right) \tag{3.7}$$

where $\phi_u$ and $\phi_p$ are the unary and pairwise potentials.

The unary potential $\phi_u$ is computed independently for each part, and is based on the visual features (*i.e.* the feature maps $z$) only. It does not consider any relations between parts nor produce consistency between their displacements. For each part $(i,j)$, it gives a negative log-probability distribution over possible displacements for that part. We use the score function $S_{i,j}$ from the independent deformation model (defined in Equation 3.1 from Section 3.3.2.1) as unnormalized probability distribution and apply a $SoftMax$ function to it to obtain a valid distribution, yielding

$$\phi_u\left(d(i,j)\right) = -LogSoftMax\left[S_{i,j}\right]\left(d(i,j)\right). \tag{3.8}$$

The main purpose of using a CRF is to use a pairwise potential $\phi_p$ to relate pairs of displacements in order to enforce consistency between them (see Figure 3.4). We use it here to smooth the deformation field over the region by introducing the constraint that nearby parts should have similar displacements, through the design of a specific form for the potential $\phi_p$. Doing so, it increases the robustness of the part alignment stage. Following (Krähenbühl et al. 2011), we use a potential of the form

$$\phi_p\left(d(i,j), d(i',j')\right) = w_0\, k\left((i,j),(i',j')\right)\, \mu\left(d(i,j), d(i',j')\right) \tag{3.9}$$

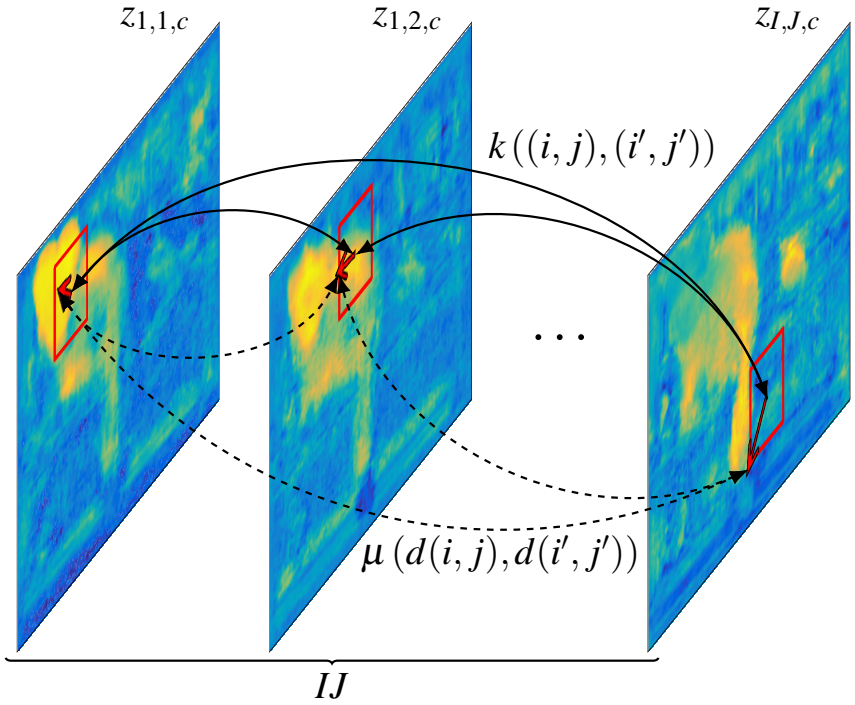



Figure 3.4 – **Visualization of pairwise potentials of CRF between parts** for a region of a class $c$. Interactions between all $IJ$ parts are taken into account through pairwise potentials $\phi_p$. These are composed of two main terms: a kernel $k$ controlling the strength of the interactions according to the distances between parts, and a compatibility function $\mu$ encouraging similarity of displacements.

where $w_0$ is the weight of the pairwise component, $k$ is a gaussian kernel and $\mu$ is a compatibility function between displacements.

We define dedicated functions $k$ and $\mu$ suited to our particular problem of computing deformations of a region. The kernel $k$ controls the weights of the pairwise links according to how far apart the parts are, and has the following expression:

$$k\left((i,j),(i',j')\right) = \exp\left(-\frac{|i-i'|^2+|j-j'|^2}{2\sigma^2}\right) \tag{3.10}$$

with $\sigma$ giving the width of the kernel. The compatibility function $\mu$ gives the penalty assigned to a pair of displacements, and we choose it so that the deformation field over the region tends to be smoother, then acting as a regularization:

$$\mu\left(d(i,j),d(i',j')\right) = \frac{|dx(i,j)-dx(i',j')|^2}{\sigma_d} + \frac{|dy(i,j)-dy(i',j')|^2}{\sigma_d} \tag{3.11}$$

with $\sigma_d$ controlling the strength of the penalty according to how similar the displacements are. Other norms can also be used in $\mu$ (*i.e.* changing the exponent

of the power), but they experimentally do not yield any improvement. In summary, the pairwise potential $\phi_p$ takes the form

$$\phi_p\left(d(i,j), d(i',j')\right) = w_p \exp\left(-\frac{|i-i'|^2 + |j-j'|^2}{2\sigma^2}\right) \times \left(|dx(i,j) - dx(i',j')|^2 + |dy(i,j) - dy(i',j')|^2\right) \quad (3.12)$$

where $w_p = \frac{w_0}{\sigma_d}$.

We run $T$ iterations of a Mean Field algorithm to perform approximate inference on the CRF, and use an efficient gaussian filtering in order to speed it up (Krähenbühl et al. 2011). This is done simultaneously for all classes $c$ and all regions $R$ at each forward pass, *i.e.* all the CRFs are optimized in parallel, in order to obtain all the deformations $d_c^R$. These are then used to backpropagate gradients at selected locations, as done with independent deformations. While there are multiple CRFs to optimize at the same time, they are all rather small since the number of variables (*i.e.* the number of parts $IJ$) is limited. Therefore, this only adds a moderate overhead compared to having independent deformations. In all experiments, we use $w_p = 0.3$, $\sigma = 1.3$ and we perform a single Mean Field iteration (*i.e.* $T = 1$), as doing more iterations does not lead to significant improvement.

We note that this CRF-based formulation of deformable part-based RoI pooling is a generalization of the independent deformation formulation (Section 3.3.2.1). Indeed, setting the pairwise weight $w_p = 0$ or doing no iteration of Mean Field inference (*i.e.* $T = 0$) results in maximizing $S_{i,j}$, which is exactly Equation 3.2.

### 3.3.3 Classification and Localization Predictions with Deformable Parts

Predictions are performed with two sibling branches, for classification and relocalization of region proposals, as is common practice (Girshick 2015). The classification branch is simply composed of an average pooling followed by a SoftMax layer. This is the strategy employed in R-FCN (Dai, Y. Li, et al. 2016), but the deformations introduced before (with deformable part-based RoI pooling) bring more invariance to transformations of objects and boost classification.

Regarding localization, the same approach is used by R-FCN, *i.e.* a simple average of pooled localization values. However, this is not adapted to DP-FCN as it is for classification, due to the presence of deformations. Indeed, while the positions and dimensions of input bounding boxes are implied by the pooling regions (*i.e.* parts) in R-FCN, it is no longer the case when those are moved by a deformable part-based RoI pooling layer. With the same strategy as R-FCN, the network would not keep track of the displacements of parts (which are never made explicit in this

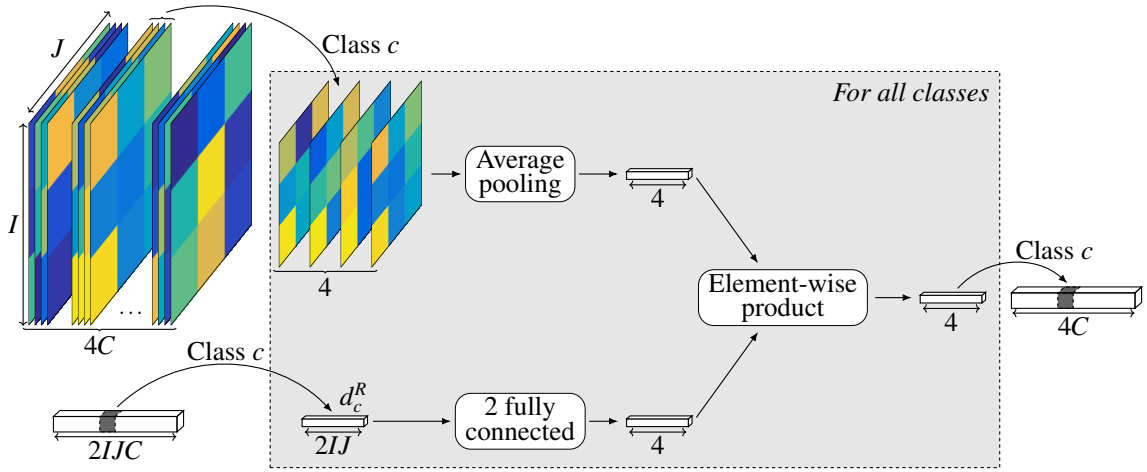



Figure 3.5 – **Deformation-aware global localization refinement.** Relocalizations of bounding boxes obtained by averaging pooled values from localization maps (upper path) do not benefit from deformable parts. To do so, displacements of parts are forwarded through two fully connected layers (lower path) and are element-wise multiplied with the previous output to refine it, separately for each class. Localization is done with 4 values per class, following (Girshick, Donahue, et al. 2014; Girshick 2015).

architecture) and would therefore be unaware of the exact input bounding box to be relocalized, leading to approximate localization.

To solve that issue, we introduce a deformation-aware localization module, explicitly taking deformations of parts into account. Since we want bounding boxes to tightly enclose objects, localization should not be invariant to local transformations but adapt accordingly. The configuration of parts (*i.e.* their positions relative to each other) is obtained as a by-product of the alignment of parts performed before, and can then be exploited to refine naive localization predictions obtained from pooling at deformed locations, so that exact geometries of bounding boxes are recovered. It also gives rich geometric information about the appearances of objects, *e.g.* their shapes or poses, that can be used to further enhance localization accuracy.

In the following sections, we introduce two versions of the localization refinement module. The first approach computes naive, deformation-unaware predictions, then uses displacements of parts to improve them. Rather than considering global predictions only, the second method exploits partial predictions made by all parts individually, and directly combines them with displacements of parts to yield final predictions. That way, interactions between both positions and outputs of all parts can be expressed, resulting in a more accurate localization.

For both modules, the refinement is mainly geometric rather than semantic, *i.e.* it depends only on the displacements of parts and not on the classes of objects. Therefore, the same configuration of parts should give the same refinement. For this reason, the localization is applied for each class separately and parameters are shared between classes. Additionally, sharing parameters can act as a regularization for classes with fewer examples.

#### 3.3.3.1 Global Localization Refinement

This localization module separately processes outputs and displacements of parts, for a class $c$ and a region $R$, before merging them with a simple operation (see Figure 3.5). It exploits the strategy of R-FCN, *i.e.* an average pooling of partial predictions from parts, to compute a first deformation-unaware prediction (upper path in Figure 3.5). This output is based on visual features only, without considering deformations, as noted before.

For that reason, we extract the feature vector $d_c^R$ of normalized displacements $(dx_c^R, dy_c^R)$ of all parts, computed by the deformable part-based RoI pooling layer (as shown in the bottom right corner of Figure 3.3), and use it to refine previous naive prediction. $d_c^R$, of size $2IJ$ (*i.e.* a 2D displacement for each part), is forwarded through a simple sub-network (lower path in Figure 3.5) to yield a feature vector of size 4 (the same as the prediction, following (Girshick, Donahue, et al. 2014; Girshick 2015)) encoding the positions of parts. The sub-network is composed of two fully connected layers with a Rectified Linear Unit (ReLU) between them. The size of the first layer is set to 256 in all our experiments. The result is then element-wise multiplied with the first prediction to adjust it accordingly to the exact locations where it was computed, yielding the final localization output.

#### 3.3.3.2 Bilinear Localization Refinement

While the previous method computes a prediction and only globally refines it with deformations, this second approach to localization refinement jointly considers all partial predictions and displacements of parts in a single operation. That way, it expresses interactions between parts more effectively and at a finer level.

To do this we use a bilinear product between predictions and displacements, that directly outputs the final localization (see Figure 3.6), which is of size 4 as before. With that operation, all pairs of prediction and displacement, even from different parts, contribute to the output. It can therefore model richer and more complex shapes than the global relocalization, and the final detections are more accurate.

To reduce computation here, we use a Tucker decomposition (Tucker 1966): we compute two feature vectors $u_c^R$ and $v_c^R$ of lower size $s$ for both partial predictions and displacements, with a simple fully connected layer applied to each input, and

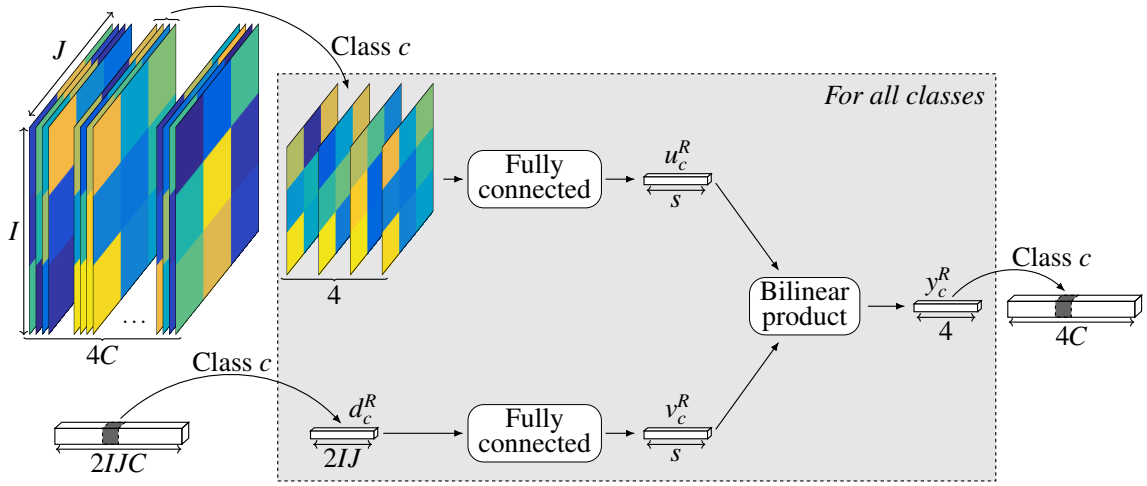



Figure 3.6 – **Deformation-aware bilinear localization refinement.** For each region and class, both predictions and displacements from all parts are separately embedded into lower dimensional features before feeding a bilinear product layer (*i.e.* a Tucker decomposition) to yield final localization prediction of size 4, following (Girshick, Donahue, et al. 2014; Girshick 2015). This kind of refinement naturally learns relations between pairs of parts, and so describes shapes of objects more finely.

only feed these two vectors into the bilinear layer. Each of the four localization output values $y_c^R$ is then obtained with

$$y_c^R(l) = \sum_{m=1}^{s} \sum_{n=1}^{s} u_c^R(m) \mathrm{T}(m,n,l) v_c^R(n) + b(l) \tag{3.13}$$

where T is a tensor of size $s \times s \times 4$ and $b$ is a bias of size 4, both learned within the layer and shared between classes. In all experiments, we use a reduced size of $s = 32$, which keeps memory and computation requirements low. While having bigger features yields slightly better results, we think this is a good trade-off between performance and computation. More complex combination operations could be used instead of the Tucker decomposition to further improve performance, *e.g.* MUTAN (Ben-Younes et al. 2017).

## 3.4 Experiments

### 3.4.1 Main Results

**Experimental setup.** We perform this analysis with the fully convolutional backbone architecture ResNet-50 (He, X. Zhang, et al. 2016a) whose model, pre-

trained on ImageNet (Russakovsky et al. 2015), is freely available. The network is trained with Stochastic Gradient Descent (SGD) for 60,000 iterations with a learning rate of $5 \cdot 10^{-4}$ and for 20,000 further iterations with $5 \cdot 10^{-5}$. The momentum parameter is set to 0.9 and the weight decay to $10^{-4}$. Each mini-batch is composed of 64 regions from a single image at the scale of 600px, selected according to Fast R-CNN (Girshick 2015). Horizontal flipping of images with probability 0.5 is used as data augmentation. We exploit the region proposals computed by AttractioNet (Gidaris et al. 2016b; Gidaris et al. 2016a) released by the authors. The top 2,000 regions are used for learning and the top 300 are evaluated during inference. We use $I \times J = 7 \times 7$ parts, as advised by the authors of R-FCN (Dai, Y. Li, et al. 2016). As is common practice, detections are post-processed with Non-Maximum Suppression (NMS) with the standard threshold of 0.3.

All experiments in this section are conducted on PASCAL VOC 07+12 dataset (Everingham et al. 2015): training is done on the union of the 2007 and 2012 trainval sets (16,551 images) and testing on the 2007 test set (4,952 images). In addition to the standard mean Average Precision (mAP)@0.5 (*i.e.* PASCAL VOC style) metric, results are also reported with the mAP@0.75 and mAP@[0.5:0.95] (*i.e.* MS COCO style) metrics to thoroughly evaluate the effects of proposed improvements.

**Performances of models.** Performance of our implementation of R-FCN (Dai, Y. Li, et al. 2016) with the given setup is shown in the first row of Table 3.1. Using independent deformations and global localization refinement, DP-FCN (second row of Table 3.1) outperforms R-FCN in all three metrics with large margins. In particular, it gains 2.0 points in mAP@0.5 over R-FCN. Then, with the improved joint deformations and bilinear localization refinement, DP-FCN2.0 (last row of Table 3.1) has better results, with an significant improvement of 4.4 points in mAP@0.75 with respect to DP-FCN. These results validate the effectiveness of deformations within networks to enhance detection, and also that richer models of deformations (*i.e.* with interactions between parts) lead to better performance.

### 3.4.2 Ablation Study

**Experimental setup.** For this ablation study, we use the same experimental setup as before (Section 3.4.1) so that results are directly comparable.

**Analysis of models.** We present a detailed analysis of results for each new module in Table 3.3 for the three metrics mAP@0.5, mAP@0.75 and mAP@[0.5:0.95]. In each table, R-FCN is shown in the top left corner as the baseline. Adding the deformable part-based RoI pooling with independent deformations to R-FCN (second rows of tables) improves mAP@0.5 by 1.7 points. Indeed, this metric is rather permissive so the localization does not need to be very accurate. On the

| Model | | | | | Results | | |
|---|---|---|---|---|---|---|---|
| Name | Independent deformations | Joint deformations | Global localization refinement | Bilinear localization refinement | mAP@ 0.5 | mAP@ 0.75 | mAP@ [0.5:0.95] |
| R-FCN (Dai, Y. Li, et al. 2016) | | | | | 74.1 | 39.4 | 40.0 |
| DP-FCN (ours) | ✓ | | ✓ | | 76.1 | 40.9 | 41.3 |
| DP-FCN2.0 (ours) | | ✓ | | ✓ | **76.5** | **45.3** | **43.2** |

Table 3.1 – **Main results of DP-FCN and DP-FCN2.0** on PASCAL VOC 2007 test in average precision (%). Without deformable part-based RoI pooling nor localization refinement module, it is equivalent to R-FCN (the reported results are those of our implementation with the given setup).

| Model | Number of parameters | Number of FLOPs | Forward time (s) |
|---|---|---|---|
| R-FCN (Dai, Y. Li, et al. 2016) | 32.26 M | 133.6 G | 0.167 |
| DP-FCN (ours) | 32.28 M | 134.3 G | 0.299 |
| DP-FCN2.0 (ours) | 32.27 M | 152.5 G | 0.492 |

Table 3.2 – **Runtime analysis of DP-FCN and DP-FCN2.0.** Values reported are computed with ResNet-50 on images at scale of 600px, and averaged over PASCAL VOC 2007 test.

| (mAP@0.5) | No localization refinement | Global localization refinement | Bilinear localization refinement |
|---|---|---|---|
| No deformation | R-FCN 74.1 | – | – |
| Independent deformations | 75.8 (+1.7) | DP-FCN 76.1 (+2.0) | 76.4 (+2.3) |
| Joint deformations | – | 76.4 (+2.3) | DP-FCN2.0 **76.5 (+2.4)** |

| (mAP@0.75) | No localization refinement | Global localization refinement | Bilinear localization refinement |
|---|---|---|---|
| No deformation | R-FCN 39.4 | – | – |
| Independent deformations | 38.8 (-0.6) | DP-FCN 40.9 (+1.5) | 45.0 (+5.6) |
| Joint deformations | – | 40.5 (+1.1) | DP-FCN2.0 **45.3 (+5.9)** |

| (mAP@[0.5:0.95]) | No localization refinement | Global localization refinement | Bilinear localization refinement |
|---|---|---|---|
| No deformation | R-FCN 40.0 | – | – |
| Independent deformations | 40.4 (+0.4) | DP-FCN 41.3 (+1.3) | 42.9 (+2.9) |
| Joint deformations | – | 41.6 (+1.6) | DP-FCN2.0 **43.2 (+3.2)** |

Table 3.3 – **Ablation study of DP-FCN and DP-FCN2.0** in mAP@0.5, mAP@0.75 and mAP@[0.5:0.95] on PASCAL VOC 2007 test in average precision (%). Results are given with absolute performances, with improvements with respect to R-FCN between parenthesis.

other hand, we see a negative effect on mAP@0.75. That is due to the uncertainty in the positions of parts, leading to an imprecise localization as already noted in Section 3.3.3. Overall, this is still beneficial, with a gain of 0.4 points in mAP@[0.5:0.95]. The improvements are therefore mainly due to a better recognition, thus validating the role of deformable parts. With the global localization refinement module (second columns of tables), the mAP@0.5 has only a small improvement, because localization accuracy is not a issue. However, it further improves mAP@0.75 by 2.1 points (*i.e.* 1.5 points with respect to R-FCN) and mAP@[0.5:0.95] by 0.9 points, validating the need for such a module. This confirms that it solves the previous issue of approximate localization and that aligning parts brings geometric information useful for localization.

We then change the independent deformations to use the joint CRF-based ones (last rows of tables), which brings an additional improvement of 0.3 points for both mAP@0.5 and mAP@[0.5:0.95] metrics with respect to DP-FCN. This therefore confirms that deformations play an important role in recognition, as already noted. When using the bilinear localization refinement (last columns of tables) in place of the global one, it yields great improvements of 4.1 and 1.6 points in mAP@0.75 and mAP@[0.5:0.95] respectively, while it is smaller in mAP@0.5. This again confirms that this module is mainly dealing with the accuracy of the localization, but not with the recognition of the object categories. By combining both improved modules (bottom right corners of tables), DP-FCN2.0 has additional gains in all three metrics, showing that the two contributions are complementary, and validates the importance of taking interactions of parts into account for accurate predictions.

### 3.4.3 Further Analysis

**Comparison with R-FCN.** Some examples of detection outputs are illustrated in Figure 3.7 to visually compare R-FCN and DP-FCN, and evaluate proposed improvements. It appears that R-FCN can more easily miss extremal parts of objects (see first two rows, *e.g.* the woman's left arm or the ears of the horse), and that DP-FCN is better at separating close instances (see last two rows, *e.g.* people or boats next to each other), thanks to deformable parts. While detections from DP-FCN and DP-FCN2.0 are often rather similar, the latter generally fits objects more tightly. We show some examples of that in Figure 3.8.

**Runtime analysis.** We present some runtime statistics about R-FCN, DP-FCN and DP-FCN2.0 in Table 3.2. The first column shows that all models have roughly the same number of parameters, *i.e.* our approaches do not bring many additional parameters and so should not need significantly more examples to be learned. The average number of FLOPs (multiply-adds) and times of network forward passes are displayed in the following two columns. It is noticeable that DP-FCN

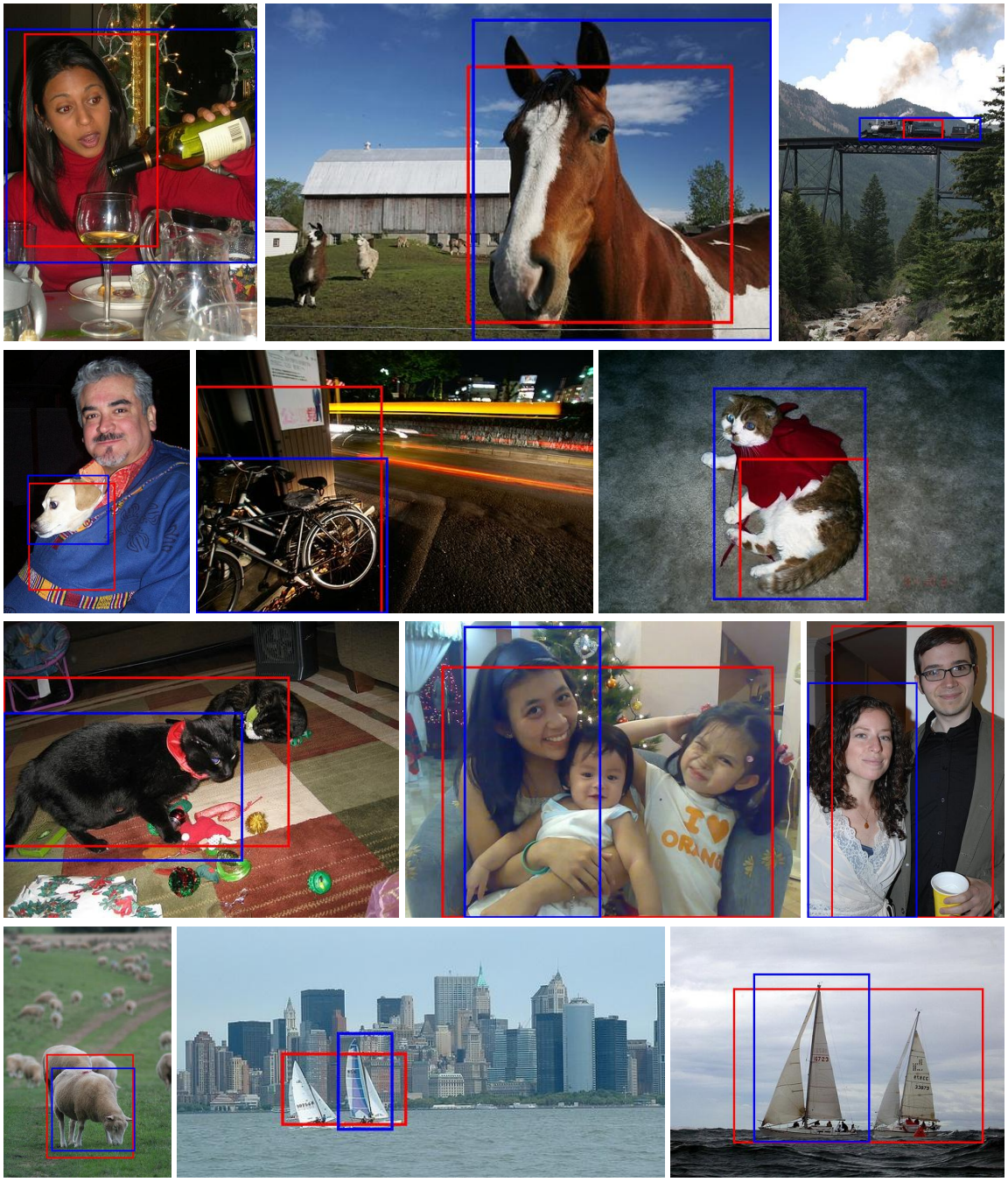


Figure 3.7 – **Comparison of detections from R-FCN (red) and DP-FCN (blue).** DP-FCN tightly fits objects (first two rows) and separates close instances (last two rows) better than R-FCN.

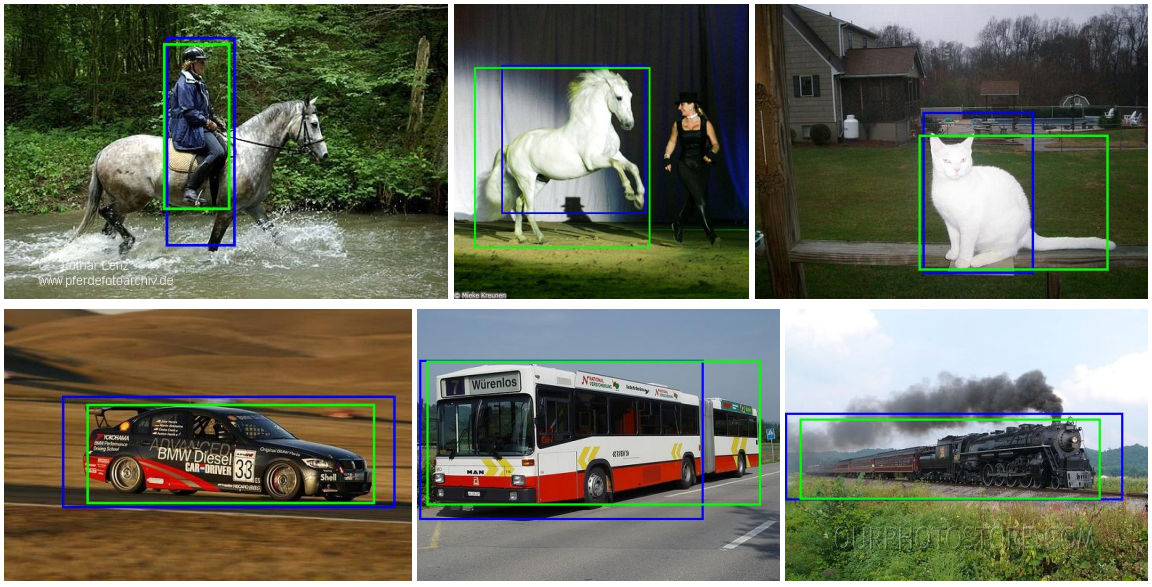



Figure 3.8 – **Comparison of detections from DP-FCN (blue) and DP-FCN2.0 (green).** Predictions of DP-FCN2.0 are better localized in general.

yields a moderate overhead compared to R-FCN, while the more computational intensive inference carried out by DP-FCN2.0, because of the CRFs introduced, leads to a heavier model.

**Interpretation of parts.** As in the original DPM (Felzenszwalb et al. 2010), the semantics of parts is not explicit in our model. Part positions are instead automatically learned to optimize detection performance, in a weakly supervised manner. Therefore the interpretation in terms of semantic parts is not systematic, especially because our division of regions into parts is finer than in DPM, leading to smaller part areas. Some deformed parts are displayed on Figure 3.9 for DP-FCN and Figure 3.10 for DP-FCN2.0, with a $3 \times 3$ part division for easier visualization. It is noticeable that the models are able to better fit to objects with deformable parts than with simple bounding boxes.

**Network architecture.** We compare DP-FCN with several FCN backbone architectures in Table 3.4, in particular the 50- and 101-layer versions of ResNet (He, X. Zhang, et al. 2016a), Wide ResNet (Zagoruyko and Komodakis 2016) and ResNeXt (Xie et al. 2017). We see that the detection mAP of DP-FCN can be significantly increased by using better networks. ResNeXt-101 (64x4d) gives the best results among the tested ones, with large improvements in all metrics, despite not using dilated convolutions. We expect DP-FCN2.0 to behave similarly, in particular to give the best results with ResNeXt-101 (64x4d) too.

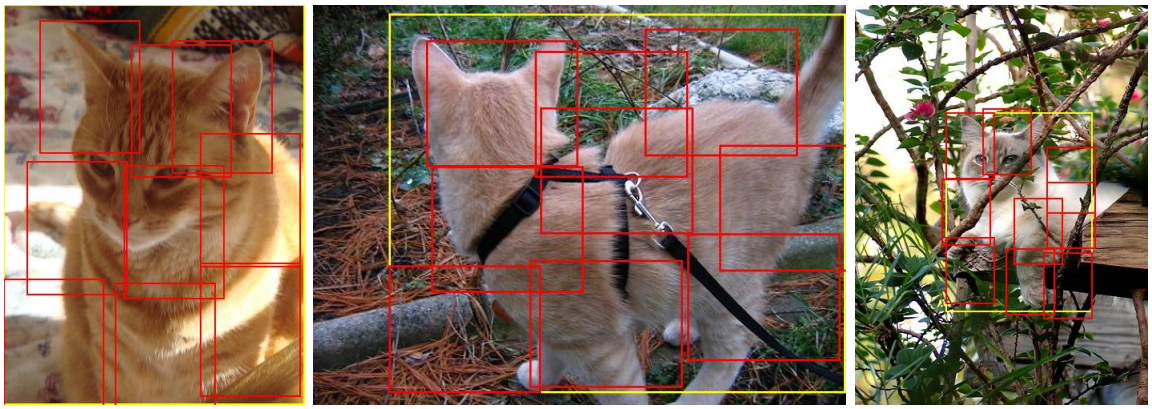


Figure 3.9 – **Examples of deformations of parts from DP-FCN.** Initial region proposals are shown in yellow and deformed parts in red. Only $3 \times 3$ parts are displayed for clarity.

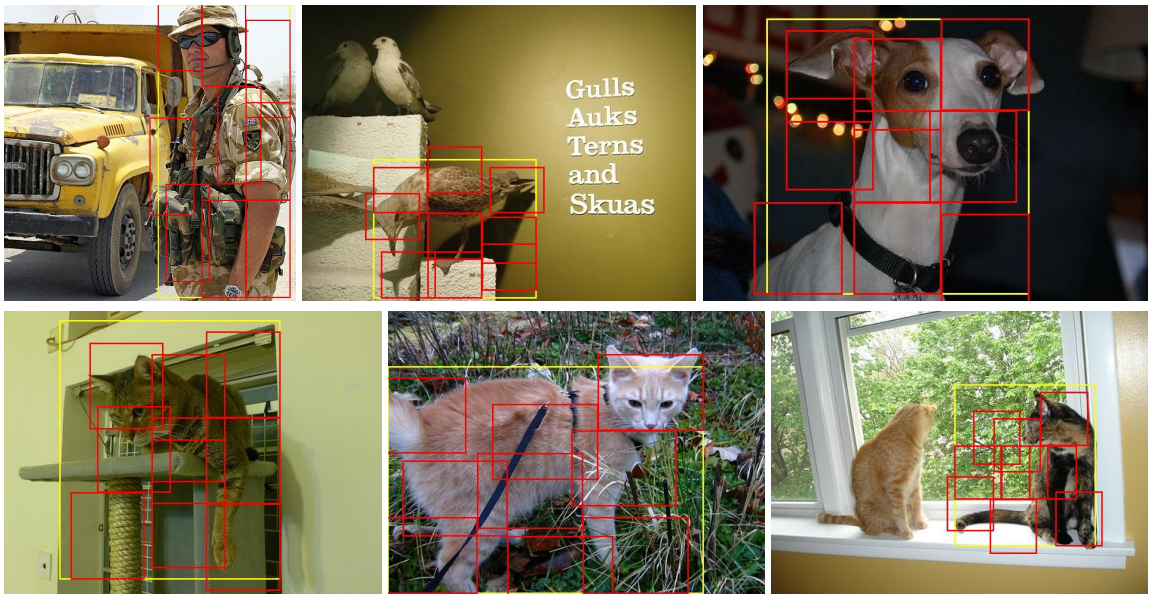



Figure 3.10 – **Examples of deformations of parts from DP-FCN2.0.** Initial region proposals are shown in yellow and deformed parts in red. Only $3 \times 3$ parts are displayed for clarity.

| FCN architecture for DP-FCN | mAP@0.5 | mAP@0.75 | mAP@[0.5:0.95] |
|---|---|---|---|
| ResNet-50 (He, X. Zhang, et al. 2016a) | 76.1 | 40.9 | 41.3 |
| ResNeXt-50 (32x4d) (Xie et al. 2017)⋆ | 76.3 | 40.8 | 41.4 |
| Wide ResNet-50-2 (Zagoruyko and Komodakis 2016) | 77.9 | 43.3 | 42.9 |
| ResNet-101 (He, X. Zhang, et al. 2016a) | 78.1 | 44.2 | 43.6 |
| ResNeXt-101 (32x4d) (Xie et al. 2017)⋆ | 78.6 | 45.2 | 44.4 |
| ResNeXt-101 (64x4d) (Xie et al. 2017)⋆ | **79.5** | **47.8** | **45.7** |

Table 3.4 – **Comparison of different FCN architectures used with DP-FCN** on PASCAL VOC 2007 test in average precision (%). Entries marked with ⋆ do not use dilated convolutions.

### 3.4.4 Comparison with State of the Art

**Experimental setup.** In order to achieve the best results possible, we bring the following improvements to the setup of Section 3.4.2: we first replace ResNet-50 by ResNeXt-101 (64x4d) (Xie et al. 2017) and increase the number of iterations to 120,000 and 40,000 on PASCAL VOC datasets, and to 480,000 and 160,000 on MS COCO dataset, with the same learning rates, using 2 images per mini-batch with the same number of regions per image. We include common tricks: color data augmentations (Krizhevsky et al. 2012), bounding box voting (Gidaris et al. 2015) with a threshold of 0.5 on PASCAL VOC and 0.75 on MS COCO, and averaging of detections between original and flipped images (Bell et al. 2016; Zagoruyko, Lerer, et al. 2016). We set the relative weight of the multi-task (classification/localization) loss (Girshick 2015) to 7 and enlarge input boxes by a factor 1.3 to include some context.

**PASCAL VOC 2007 and 2012.** Results of DP-FCN and DP-FCN2.0, along with those of recent methods, are reported in Table 3.5 for PASCAL VOC 2007 and in Table 3.6 for PASCAL VOC 2012. For fair comparisons we only report results of methods trained on VOC 07+12 (VOC07 trainval and VOC12 trainval sets) and VOC 07++12 (VOC07 trainvaltest and VOC12 trainval sets) respectively, but using additional data, *e.g.* MS COCO images, usually improves results (He, X. Zhang, et al. 2016a; Dai, Y. Li, et al. 2016). DP-FCN achieves 83.1% and 80.9% on these two datasets, yielding large gaps with all competing methods. In particular, DP-FCN outperforms R-FCN (Dai, Y. Li, et al. 2016) by significant margins (2.6 and 3.3 points respectively). DP-FCN2.0 yields 83.3% and 81.2% on PASCAL VOC 2007 and 2012 respectively, which are small additional improvements of 0.2 and 0.3 points with respect to DP-FCN. As studied in Section 3.4.2, the main improvement of this model lies in the accuracy of localization, which is not reflected here with the official PASCAL VOC metric, *i.e.* mAP@0.5. We note that these results could be

| Method | mAP | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fast R-CNN (Girshick 2015) | 70.0 | 77.0 | 78.1 | 69.3 | 59.4 | 38.3 | 81.6 | 78.6 | 86.7 | 42.8 | 78.8 | 68.9 | 84.7 | 82.0 | 76.6 | 69.9 | 31.8 | 70.1 | 74.8 | 80.4 | 70.4 |
| HyperNet (Kong et al. 2016) | 76.3 | 77.4 | 83.3 | 75.0 | 69.1 | 62.4 | 83.1 | 87.4 | 87.4 | 57.1 | 79.8 | 71.4 | 85.1 | 85.1 | 80.0 | 79.1 | 51.2 | 79.1 | 75.7 | 80.9 | 76.5 |
| Faster R-CNN (S. Ren et al. 2015) | 76.4 | 79.8 | 80.7 | 76.2 | 68.3 | 55.9 | 85.1 | 85.3 | 89.8 | 56.7 | 87.8 | 69.4 | 88.3 | 88.9 | 80.9 | 78.4 | 41.7 | 78.6 | 79.8 | 85.3 | 72.0 |
| SSD (W. Liu et al. 2016) | 76.8 | 82.4 | 84.7 | 78.4 | 73.8 | 53.2 | 86.2 | 87.5 | 86.0 | 57.8 | 83.1 | 70.2 | 84.9 | 85.2 | 83.9 | 79.7 | 50.3 | 77.9 | 73.9 | 82.5 | 75.3 |
| MR-CNN (Gidaris et al. 2015) | 78.2 | 80.3 | 84.1 | 78.5 | 70.8 | 68.5 | 88.0 | 85.9 | 87.8 | 60.3 | 85.2 | 73.7 | 87.2 | 86.5 | 85.0 | 76.4 | 48.5 | 76.3 | 75.5 | 85.0 | 81.0 |
| LocNet (Gidaris et al. 2016b) | 78.4 | 80.4 | 85.5 | 77.6 | 72.9 | 62.2 | 86.8 | 87.5 | 88.6 | 61.3 | 86.0 | 73.9 | 86.1 | 87.0 | 82.6 | 79.1 | 51.7 | 79.4 | 75.2 | 86.6 | 77.7 |
| FRCN OHEM (Abhinav Shrivastava et al. 2016) | 78.9 | 80.6 | 85.7 | 79.8 | 69.9 | 60.8 | 88.3 | 87.9 | 89.6 | 59.7 | 85.1 | **76.5** | 87.1 | 87.3 | 82.4 | 78.8 | 53.7 | 80.5 | 78.7 | 84.5 | 80.7 |
| ION (Bell et al. 2016) | 79.4 | 82.5 | 86.2 | 79.9 | 71.3 | 67.2 | 88.6 | 87.5 | 88.7 | 60.8 | 84.7 | 72.3 | 87.6 | 87.7 | 83.6 | 82.1 | 53.8 | 81.9 | 74.9 | 85.8 | 81.2 |
| R-FCN (Dai, Y. Li, et al. 2016) | 80.5 | 79.9 | 87.2 | 81.5 | 72.0 | 69.8 | 86.8 | 88.5 | 89.8 | **67.0** | **88.1** | 74.5 | 89.8 | 90.6 | 79.9 | 81.2 | 53.7 | 81.8 | **81.5** | 85.9 | 79.9 |
| DP-FCN (ours) | 83.1 | 89.8 | **88.6** | **85.2** | 73.9 | **74.7** | **92.1** | 90.4 | **94.4** | 58.3 | 84.9 | 75.2 | 93.4 | **93.1** | **87.4** | **85.9** | 53.9 | 85.3 | 80.0 | 90.4 | 85.9 |
| DP-FCN2.0 (ours) | **83.3** | **92.0** | **88.6** | 83.9 | **75.9** | 72.8 | 89.9 | **91.5** | 93.1 | 57.4 | 85.5 | 75.4 | **94.1** | 92.7 | 87.0 | 85.0 | **55.6** | **85.6** | 80.6 | **92.7** | **86.2** |

Table 3.5 – **Detailed detection results on PASCAL VOC 2007 test** in average precision (%). For fair comparisons, the table only includes methods trained on PASCAL VOC 07+12.

| Method | mAP | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fast R-CNN (Girshick 2015) | 68.4 | 82.3 | 78.4 | 70.8 | 52.3 | 38.7 | 77.8 | 71.6 | 89.3 | 44.2 | 73.0 | 55.0 | 87.5 | 80.5 | 80.8 | 72.0 | 35.1 | 68.3 | 65.7 | 80.4 | 64.2 |
| HyperNet (Kong et al. 2016) | 71.4 | 84.2 | 78.5 | 73.6 | 55.6 | 53.7 | 78.7 | 79.8 | 87.7 | 49.6 | 74.9 | 52.1 | 86.0 | 81.7 | 83.3 | 81.8 | 48.6 | 73.5 | 59.4 | 79.9 | 65.7 |
| Faster R-CNN (S. Ren et al. 2015) | 73.8 | 86.5 | 81.6 | 77.2 | 58.0 | 51.0 | 78.6 | 76.6 | 93.2 | 48.6 | 80.4 | 59.0 | 92.1 | 85.3 | 84.8 | 80.7 | 48.1 | 77.3 | 66.5 | 84.7 | 65.6 |
| SSD (W. Liu et al. 2016) | 74.9 | 87.4 | 82.3 | 75.8 | 59.0 | 52.6 | 81.7 | 81.5 | 90.0 | 55.4 | 79.0 | 59.8 | 88.4 | 84.3 | 84.7 | 83.3 | 50.2 | 78.0 | 66.3 | 86.3 | 72.0 |
| FRCN OHEM (Abhinav Shrivastava et al. 2016) | 76.3 | 86.3 | 85.0 | 77.0 | 60.9 | 59.3 | 81.9 | 81.1 | 91.9 | 55.8 | 80.6 | **63.0** | 90.8 | 85.1 | 85.3 | 80.7 | 54.9 | 78.3 | 70.8 | 82.8 | 74.9 |
| ION (Bell et al. 2016) | 76.4 | 88.0 | 84.6 | 77.7 | 63.7 | 63.6 | 80.8 | 80.8 | 90.9 | 55.5 | 81.9 | 60.9 | 89.1 | 84.9 | 84.2 | 83.9 | 53.2 | 79.8 | 67.4 | 84.4 | 72.9 |
| R-FCN (Dai, Y. Li, et al. 2016) | 77.6 | 86.9 | 83.4 | 81.5 | 63.8 | 62.4 | 81.6 | 81.1 | 93.1 | 58.0 | 83.8 | 60.8 | 92.7 | 86.0 | 84.6 | 84.4 | 59.0 | 80.8 | 68.6 | 86.1 | 72.9 |
| DP-FCN (ours)[a] | 80.9 | 89.3 | 84.2 | **85.4** | **74.4** | 70.0 | 84.0 | **86.2** | 93.9 | **62.9** | 85.1 | 62.7 | 92.7 | 87.4 | 86.0 | 86.8 | **61.3** | 85.1 | **74.8** | **88.2** | **78.5** |
| DP-FCN2.0 (ours)[b] | **81.2** | **89.8** | **85.6** | 84.7 | 74.3 | **70.8** | **85.1** | 85.4 | **94.3** | 62.6 | **86.5** | 62.0 | **92.8** | **88.4** | **88.0** | **87.4** | 61.0 | **85.4** | 73.7 | 88.0 | 78.3 |

Table 3.6 – **Detailed detection results on PASCAL VOC 2012 test** in average precision (%). For fair comparisons, the table only includes methods trained on PASCAL VOC 07++12.

*a*. http://host.robots.ox.ac.uk:8080/anonymous/QNUYVS.html
*b*. http://host.robots.ox.ac.uk:8080/anonymous/07DMTQ.html

| Method | mAP@ [0.5:0.95] | mAP@ 0.5 | mAP@ 0.75 | mAP@ Small | mAP@ Medium | mAP@ Large |
|---|---|---|---|---|---|---|
| MultiPath (Zagoruyko, Lerer, et al. 2016) (on val) | 31.5 | 49.6 | | | | |
| R-FCN (Dai, Y. Li, et al. 2016) | 31.5 | 53.2 | | 14.3 | 35.5 | 44.2 |
| ION (Bell et al. 2016) | 33.1 | 55.7 | 34.6 | 14.5 | 35.2 | 47.2 |
| DP-FCN (ours) | 34.0 | 54.7 | 37.2 | 15.9 | 36.4 | 47.5 |
| DP-FCN2.0 (ours) | 34.8 | 54.8 | 38.4 | 15.8 | 37.2 | 49.0 |
| FPN (T.-Y. Lin, Dollár, et al. 2017) | 36.2 | **59.1** | | 18.2 | 39.0 | 48.2 |
| Deformable ConvNet (Dai, Qi, et al. 2017) | 37.5 | 58.0 | | 19.4 | 40.1 | **52.5** |
| RetinaNet (T.-Y. Lin, Goyal, et al. 2017) | **39.1** | **59.1** | **42.3** | **21.8** | **42.7** | 50.2 |

Table 3.7 – **Detection results on MS COCO test-dev** in average precision (%). All methods are trained on the bounding box detection trainval set (except MultiPath which is trained on the 115k train set) and are single model.

further improved with additional common enhancements, *e.g.* multi-scale training and testing (He, X. Zhang, et al. 2015) or OHEM (Abhinav Shrivastava et al. 2016).

**MS COCO.** In order to validate the effectiveness of deformations for object detection, we present the results of DP-FCN, DP-FCN2.0 and other concurrent methods on MS COCO dataset (T.-Y. Lin, Maire, et al. 2014) in Table 3.7. It is a large-scale dataset, containing around 120,000 training images, and is more challenging than PASCAL VOC since objects are smaller in average and more numerous, effectively making the object detection task harder. While more recent approaches, *e.g.* Feature Pyramid Network (FPN) (T.-Y. Lin, Dollár, et al. 2017), RetinaNet (T.-Y. Lin, Goyal, et al. 2017), have better results, we see that DP-FCN is still competitive with the state of the art, showing the generality of our approach. It notably outperforms R-FCN again on this dataset. Again, DP-FCN2.0 yields better results than DP-FCN, with improvements of 0.8 and 1.2 points in the official and mAP@0.75 metrics, which are strict in localization. However, training on this dataset is rather computational expensive, and all the leading methods use heavy GPU resources for that. It allows them to be parameterized directly on MS COCO, while we do it on PASCAL VOC and then transfer selected values, which might be suboptimal. By training longer, tuning hyper-parameters more carefully or by integrating our ideas into newer architectures, *e.g.* FPN (T.-Y. Lin, Dollár, et al. 2017), we expect higher results.

### 3.4.5 Examples of Detections

Some example detections of the final DP-FCN model trained on VOC 07+12 data (Section 3.4.4) on unseen PASCAL VOC 2007 test images are shown in Figure 3.11 and Figure 3.12. We note that DP-FCN can successfully detect objects under simple as well as challenging conditions. The last row of Figure 3.12 shows some failure

cases where some objects are misclassified, although they are accurately localized. Example detections are illustrated in the same way for DP-FCN2.0 in Figure 3.13 and Figure 3.14.

## 3.5 Conclusion

This chapter has presented DP-FCN, a deep region-based object detector learning latent deformable part-based representations from bounding box annotations. Deformations make it more flexible than traditional region-based detectors, restricted to extract features from rectangular bounding boxes only. Using a FCN backbone architecture, deep ConvNets fully benefit from part-based representations. Parts are aligned by a deformable part-based RoI pooling layer, a MIL extension of the position-sensitive version from R-FCN (Dai, Y. Li, et al. 2016) with ideas from DPM (Felzenszwalb et al. 2010). This builds invariance to local transformations, improving recognition, and yields configurations of parts, leveraged by a deformation-aware localization module to refine bounding box regression. A further extension, DP-FCN2.0, makes interactions between parts explicit to get a finer representation. This is done by casting alignment of parts as an in-network CRF inference with custom potentials, optimizing all parts jointly, and by using a bilinear deformation-based refinement for localization.

Experiments have shown that deformable part-based representations are beneficial for object detection, both for object recognition and bounding box localization. However, comparison with recent state-of-the-art approaches on MS COCO dataset, *e.g.* FPN or RetinaNet, suggests that resolution of feature maps is more important for accurate detection. Indeed, MS COCO contains lots of rather small objects, which require enough resolution to be detected effectively. Additional experiments combining DP-FCN with more adapted backbone architectures such as FPN would be an interesting direction to explore to further improve results and compete with newest models.

The next chapter addresses the problem of object detection with additional supervision, in the form of auxiliary annotations, *i.e.* not directly related to object detection. The main question deals with leveraging this supplementary supervision to improve representation learning for the main task.


Figure 3.11 – **Example detections of** DP-FCN trained on VOC 07+12 data (Section 3.4.4) on unseen PASCAL VOC 2007 test images, using VOC color code for classes. All detections with scores above 0.6 are shown.

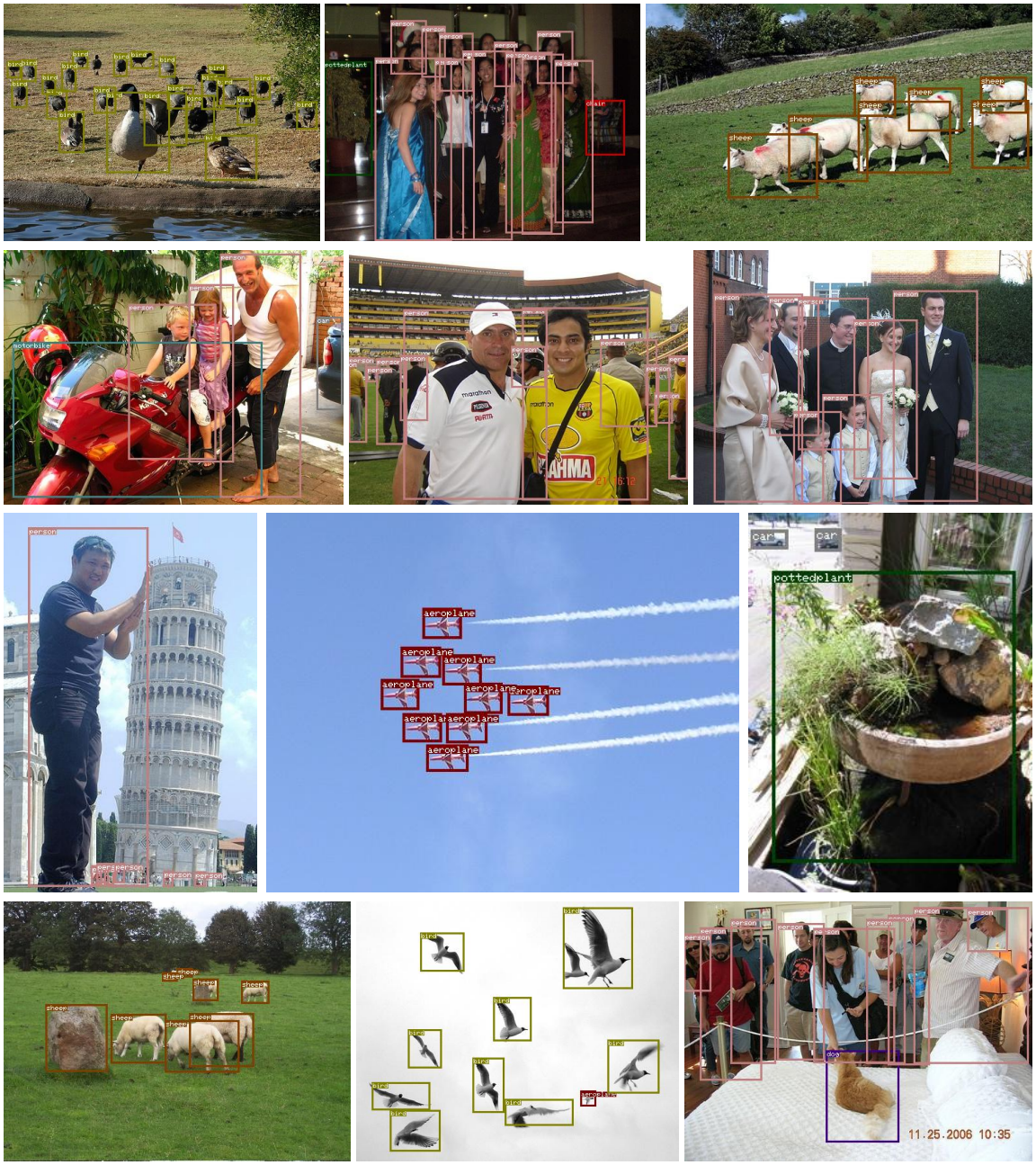



Figure 3.12 – **Example detections of DP-FCN** trained on VOC 07+12 data (Section 3.4.4) on unseen PASCAL VOC 2007 test images, using VOC color code for classes. Last row shows some failure cases. All detections with scores above 0.6 are shown.


Figure 3.13 – **Example detections of DP-FCN2.0** trained on VOC 07+12 data (Section 3.4.4) on unseen PASCAL VOC 2007 test images, using VOC color code for classes. All detections with scores above 0.6 are shown.


Figure 3.14 – **Example detections of DP-FCN2.0** trained on VOC 07+12 data (Section 3.4.4) on unseen PASCAL VOC 2007 test images, using VOC color code for classes. Last row shows some failure cases. All detections with scores above 0.6 are shown.

CHAPTER 4

# LEARNING TASK-RELATED REPRESENTATIONS FROM AUXILIARY SUPERVISIONS

## Contents




***Chapter abstract***

*This chapter deals with object detection,* i.e. *the task of both recognizing all objects from a predefined set of classes and localizing them within images with bounding boxes. Here, we assume a particularity compared to the standard setup, that more supervision is available to learn from, in addition to object boxes. We specifically consider auxiliary annotations,* i.e. *which are not directly related to object detection. While not directly solving the task, they should bring supervision of different natures. The goal is then to leverage this auxiliary supervision to improve results on object detection, the only task in which we are interested. This setup of exploiting more annotations is in particular thought as an alternative to getting more data, for learning problems with few training images or for applications where images are difficult to collect. We formalize this problem as a specific kind of Multi-Task Learning (MTL), named primary MTL, where auxiliary annotations are used to learn associated tasks, but where one task is favored, here object detection. We introduce ROCK*

*to address it. It is a generic multi-modal fusion block integrating auxiliary information into the main representation. For this, it learns to predict auxiliary annotations, and leverages associated features to fuse them into the main one to refine it. ROCK is applied with multiple geometric (depth and surface normal estimation) and semantic (scene classification) tasks as auxiliary supervision.*

*The work in this chapter has led to the publication of a conference paper:*

- Taylor Mordan, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018b). "Revisiting Multi-Task Learning with ROCK: a Deep Residual Auxiliary Block for Visual Detection". In: *Advances in Neural Information Processing Systems (NIPS)*.

## 4.1 Introduction

The outstanding success of Convolutional Neural Networks (ConvNets) for Computer Vision (CV) is due in part to the availability of large-scale annotated datasets, which has enabled training very deep models. However, training such big ConvNets is not viable when dealing with smaller-scale datasets, due to strong overfitting issues as already noted in Section 1.2.2. Even though pre-training the networks on ImageNet and transferring them helps mitigating this issue, the lack of training data still limits performances in these situations.

In some applications, *e.g.* in medical or military domains, it can be hard to collect enough images to train models properly for various reasons. This is in particular the case in the industrial context of this thesis, as explained in Section 1.1.2, due to confidentiality issues around the activities of Thales. However, in this case as in other applications, some additional annotations can be available, often provided by the global systems running the networks. Exploiting more supervision to train the models should help fighting overfitting, and yield better results. This is the idea behind Multi-Task Learning (MTL) (Caruana 1997), a common framework to exploit multiple supervisions for training a single model, and is described in Section 1.2.5. However, MTL approaches usually assume a flat structure between tasks, *i.e.* they are all equally important. The aim is then to optimize the average performance on all tasks simultaneously. This is not what we want to achieve here, where only one of the task is of interest. Indeed, this would be intrinsically sub-optimal since the problem is biased toward a given application. In this sense, our context is related to Learning Under Privileged Information (LUPI) framework (Vapnik and Vashist 2009), with auxiliary supervision being the privileged information, *i.e.* only available during training.

In this chapter, we explore leveraging additional supervision as an alternative to having more training images, in order to get more data to feed the networks and improve results on the main task, in only which we are interested. We

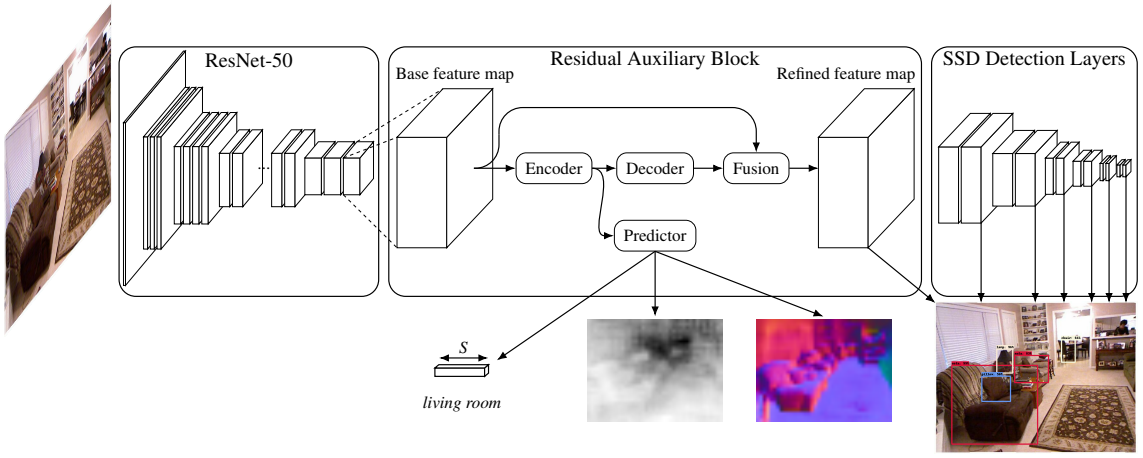



Figure 4.1 – **ROCK for object detection with auxiliary information.** ROCK (middle) is incorporated into a backbone SSD object detection model (W. Liu et al. 2016) to utilize additional supervision from multi-modal auxiliary tasks (scene classification, depth prediction and surface normal estimation) and to improve performance on the primary object detection task.

assume a specific setup, where additional annotations are auxiliary, *i.e.* they are not directly related to the main task but bring information of different natures. We use MTL to address the problem, but frame it as a new kind of MTL, named primary MTL, where a primary task is augmented during training with several auxiliary tasks leveraged to better learn it. We then introduce Residual auxiliary blOCK (ROCK) to solve it. It is a generic block that can easily be inserted into any existing architecture to effectively exploit additional auxiliary annotations. The main goal is to produce predictions for auxiliary tasks and to learn features through MTL, as illustrated in Figure 4.1. However, it is designed around two key features differentiating it from flat MTL in order to better fit ROCK to our context. First, the block is equipped with a residual connection, which explicitly merges intermediate representations of the primary and auxiliary tasks, making the latter ones have a real effect on the former in the forward pass, not just through shared feature learning as in classical MTL. Then, all layers exclusive to auxiliary predictions contain no parameters. This forces the model to learn relevant auxiliary features earlier in the intermediate representations, so as to maximize their influence on the primary task when they are fused into it. We instantiate this problem for object detection task, as in Chapter 3, with semantic (scene labels) and geometric (depth and surface normals) auxiliary annotations, but other kinds of annotations should work equally. Since we address learning problems with fewer training data, we focus on single-shot object detectors, *e.g.* Single Shot Detector (SSD) (W. Liu et al. 2016), as these are usually lighter and easier to train than region-based ones. More details are given in Section 1.2.3.

## 4.2 Related Work

MTL (Caruana 1997) is the common framework to learn from multiple supervisions. Extensive works have adapted it to the Deep Learning (DL) case (Bilen et al. 2016a; Kokkinos 2017; Meyerson et al. 2018), however they always assume a flat structure between tasks, with the goal of performing well on average across tasks, as noted in Section 1.3. Instead, we focus on primary MTL, where only one task matters. Several attempts have been made to estimate how much tasks transfer to others (Azizpour, Razavian, et al. 2016; Zamir et al. 2018), which could be useful to select what annotations to use, but we are here interested in a way to effectively leverage auxiliary supervision to improve the results on primary task.

**Learning Under Privileged Information framework.** LUPI framework (Vapnik and Vashist 2009; Pechyony et al. 2010; Vapnik and Izmailov 2015) addresses situations where additional information is available during training only, and several works have proposed ways to handle it. Seminal works on this topic use modified versions of Support Vector Machines (SVMs) to include privileged information into the learning of the classifier. SVM+ (Vapnik and Vashist 2009) and Margin Transfer (Sharmanska et al. 2013; Sharmanska et al. 2014) both estimate difficulties of examples using privileged information, to weight them accordingly during training. Another approach is Generalized Distillation (Lopez-Paz et al. 2016), which learns two models on standard and privileged data, with similarity constraints between them. In our setup, auxiliary annotations are not directly related to the primary task and can be of any form, *e.g.* a single scene class label, so these approaches are not always applicable.

LUPI framework has also been extended to DL (Hoffman et al. 2016; Shi et al. 2017). In particular for object detection, modality hallucination network (Hoffman et al. 2016) is closely connected to our method, as it uses depth as privileged information to improve object detection. However, both methods differ in the way they use the depth annotations. While modality hallucination network directly uses depth to perform object detection, our approach merges intermediate representations used for depth prediction. This difference leads to an earlier fusion in ROCK, where intermediate representations from all tasks are fused together to benefit from the correlation between tasks, while modality hallucination network uses a late fusion of predictions.

**Object detection with multi-modal data.** The use of annotations from different but related tasks to improve performance is a common approach. Our problem is related to the use of semantic (scene) and geometric (depth and surface normals) information, which have been successfully combined (B. Liu et al. 2010; Ladicky et al. 2014; P. Wang et al. 2015b; Hane et al. 2015; Eigen et al. 2015; Dharmasiri et al. 2017), although not directly for object detection and primary MTL. In the

context of object detection, combining several additional informations, *e.g.* depth (Spinello et al. 2012; Gupta, Girshick, et al. 2014; Gupta, Arbeláez, et al. 2015; C. Wang et al. 2016; Park et al. 2017), or surface normal and surface curvature (C. Wang et al. 2016) has shown to significantly improve performance. How to perform the fusion of features from different modalities so as to fully benefit from their complementarity remains an active topic, especially between color and depth domains (Spinello et al. 2012; Guerry et al. 2017; Park et al. 2017; Hazirbas et al. 2016; J. Wang et al. 2016). Our context is different since we only use auxiliary information during training, which is a less constraining framework.

More related to our context, (Z. Ren et al. 2018) leverages depth, surface normals and instance contours to pre-train a model on synthetic data through flat MTL, then fine-tunes it for object detection on real target data. We differ from this kind of approach by our MTL strategy, which is driven by an object detection primary task.

## 4.3 ROCK: Residual Auxiliary Block

The general architecture of our model is shown in Figure 4.1. It is created from an existing model performing a given task $t_0$. This model should be composed of a backbone network yielding a base feature map $X$ (left of Figure 4.1), used as input to a task-specific module computing predictions (right of Figure 4.1). This kind of design is fairly general, so this assumption is not restrictive. The idea behind ROCK is to add a new residual auxiliary block (middle of Figure 4.1) between the two existing components, in order to leverage $T$ other, auxiliary tasks $\{t_i\}_{i=1}^{T}$ to extract useful information and inject it into the base feature map $X$ to yield a refined version $\tilde{X}$ of it. This refined representation, being similar to the base feature map, is then used by the task-specific module of the primary task, which is now explicitly influenced by auxiliary tasks. The new task-specific features might not be easily learned from the primary task $t_0$ only, so $\tilde{X}$ encodes additional details of the scenes learned by the block, therefore leading to better performance on the primary task $t_0$.

To refine the base feature map $X$, the auxiliary block must extract information from all auxiliary tasks $\{t_i\}_{i=1}^{T}$. To this end, it is learned within MTL framework: during training, a prediction $y_t$ is produced for every task $t$ and a loss $\ell_t$ is applied, so that the block is learned from all tasks (including the main primary task, through the refinement path) simultaneously. Learned intermediate features are then used in the refinement step. In the inference phase, features are extracted for auxiliary tasks and are used to modify the base feature map in the same way, so that the predictions for the primary task explicitly take this information into account, without needing any annotations. Therefore, ROCK uses auxiliary supervision as privileged information.

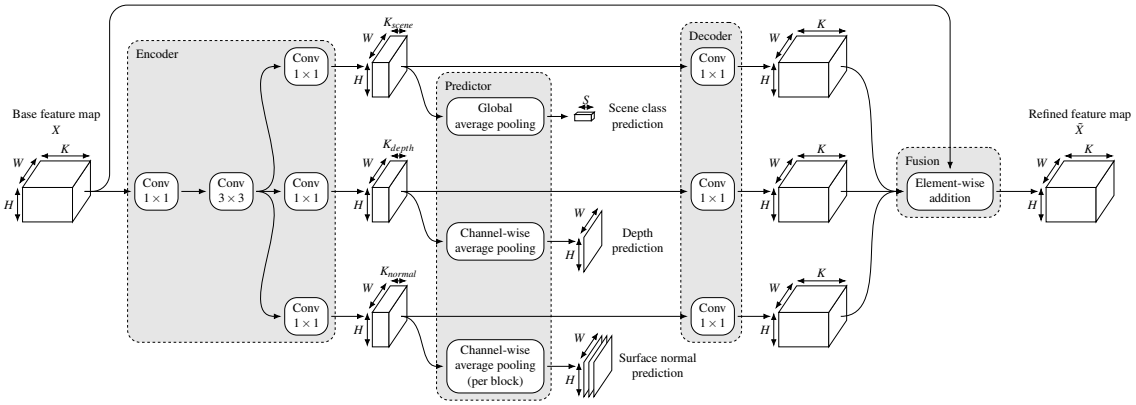



Figure 4.2 – **ROCK architecture.** The block is composed of four parts, represented by the shaded areas: the encoder extracts task-specific features for all auxiliary tasks; the decoder and fusion operation transform these encodings back to the original feature space and merge them into the main path, explicitly bringing complementary information to the primary task; the predictor produces outputs for auxiliary tasks in order to learn from them through MTL. Although the block can be instantiated for any number and kind of tasks, it is presented here with the specific setup of three auxiliary tasks described in Section 4.4.

We now present the general design of the residual auxiliary block for arbitrary tasks, then detail the architecture we use in the experiments on NYUv2 dataset with the associated tasks in Section 4.4. The block is thought to be generic, so that it can be easily integrated into a wide range of networks and can be applied to almost any task, without further major change. All its components are designed to have a small computational overhead, in order to keep the increase in complexity light, easing the integration of the block into existing architectures. It also has as few parameters as possible. The resulting model can therefore be learned efficiently, and fully leverage additional annotations to effectively increase performance.

Our auxiliary block is composed of four main parts: encoder, decoder, fusion and predictor. They are all detailed in Figure 4.2 within shaded blocks. We note that we use a simple design here to have a generic approach and show its benefits, but more complex architectures could lead to better results through better feature learning.

The base feature map $X$ is first processed by the encoder $Enc$, whose role is to learn task-specific features $\{Enc_{t_i}(X)\}_{i=1}^{T}$ from it with dedicated heads. For each task $t$, we use a bottleneck-like architecture to keep computation low. As shown in Figure 4.2, it is composed of a $1 \times 1$ convolution to reduce width of the base feature map by a factor of 4, followed by a $3 \times 3$ convolution with the same width. The last operation is a task-specific $1 \times 1$ convolution to yield a width $K_t$ adapted

to the task $t$. When learning from multiple auxiliary tasks, the first two layers of the encoder are shared to have a common encoder trunk with task-specific heads, further reducing computation. The task-specific encodings obtained here are then used as input for both the decoder and the predictor, and should therefore contain all necessary information about auxiliary tasks to be used in the refinement step. We detail how this is achieved in the following.

### 4.3.1 Merging of Primary and Auxiliary Representations

Different tasks bringing different kinds of supervision, the previous encodings $\{Enc_{t_i}(X)\}_{i=1}^{T}$ should contain information complementary to what is learned from the primary task $t_0$. Therefore, it seems useful to combine them with the base feature maps $X$ to get more complete representations of the scenes. The second and third parts of the residual auxiliary block are the decoder $Dec$ and the fusion step $F$. The former takes the output of the encoder and projects it back to the space of the base feature map, so that it can be injected back into the primary path. This is done for each task $t$ separately with a single $1 \times 1$ convolution to have the same width as the base feature map (see Figure 4.2). The fusion step then merges all these task-specific features with the base feature map in a residual manner to yield the refined feature map $\tilde{X}$, which encodes both primary and auxiliary information:

$$\tilde{X} = F\left(X, \{Dec_{t_i} \circ Enc_{t_i}(X)\}_{i=1}^{T}\right) = X + \sum_{i=1}^{T} Enc_{t_i}(X). \tag{4.1}$$

The residual formulation allows the base feature map to keep its content while focusing it more on relevant details of the images, yielding better features for the primary task. This feature merging step is key in ROCK to improve upon flat MTL, and these two modules are the main difference between flat and primary MTL. In flat MTL, all tasks are at the same level and are able to benefit each other through implicit common feature learning only, *i.e.* their mutual influence is implicit in the models. By injecting the auxiliary representations into the primary one, we break the symmetry between tasks, effectively favoring the primary one. This task is then explicitly influenced by the auxiliary tasks through fusion, and the model can fully leverage auxiliary supervision.

### 4.3.2 Effective MTL from Auxiliary Supervision

The last element of the residual auxiliary block is the predictor $Pred$. Its purpose is to produce predictions $\{y_{t_i}\}_{i=1}^{T}$ for all auxiliary tasks $\{t_i\}_{i=1}^{T}$, so that losses can be applied to learn from the tasks through MTL. Its inputs are the feature encodings $\{Enc_{t_i}(X)\}_{i=1}^{T}$ from the encoder, whose sizes are already task-specific. Since the predictor might lose information with respect to these features

in order to yield the predictions, only the features from the encoder go through the decoder to be merged back into the main path (as illustrated in Figure 4.2 and formalized in Equation 4.1), so that more information is kept for use in the primary task. Therefore, all parameters learned in the predictor are to be thrown away after training, *i.e.* they are not used for inference (we are only interested in the primary task, the auxiliary tasks being used only to improve its performance). In order to force the model to learn useful information in the encoder and not in the predictor, so that it is kept and merged back, we use a predictor composed of pooling layers only, with no learned parameter:

$$y_t = Pred_t\left(Enc_t(X)\right) = Pool_t\left(Enc_t(X)\right) \tag{4.2}$$

with $Pool_t$ a task-specific pooling operation. The kinds of pooling used are dependent on the tasks considered, as they are directly linked to the natures of the tasks (*e.g.* scalar or spatial). Once again, this design choice of not having any learned parameter in the predictor is important for ROCK to distinguish from flat MTL. It forces the task-specific representation learning to happen within the encoder, and therefore to take part in the refinement step. This is a way to maximize the influence of the auxiliary tasks on the primary one, *i.e.* to get away from flat MTL.

## 4.4 Application to Object Detection with Multi-Modal Auxiliary Information

We instantiate ROCK for object detection as the primary task, using multi-modal auxiliary information: scene classification, depth prediction and surface normal estimation, see Figure 4.1.

**Scene classification.** For scene classification, the encoder and predictor follow the common design for classification problems: the last layer of the encoder is a classification layer into $K_{scene} = S = 27$ scene classes, and the pooling is a global average pooling, reducing spatial dimensions to a single neuron while keeping width equal to the number of classes. Error is computed with a cross-entropy loss preceded by a SoftMax layer over the $S$ classes as is common in classification tasks.

**Depth estimation.** For depth estimation, annotations consist in a single depth map for each example. We choose $K_{detph}$ by dividing previous width by a factor of 4, as a trade-off between compressing maps to the final target width of 1 and keeping enough information to provide to decoder. We then use a channel-wise average pooling reducing width from $K_{depth}$ to 1, while keeping spatial dimensions. The spatial resolution of predictions is the same as that of the base feature map,

and therefore depends on where the block is inserted into the network. The regression loss used here is a reverse Huber loss (Laina et al. 2016) in log space, as it has been shown to yield good results for depth prediction.

**Surface normal estimation.** The last surface normal estimation task is similar to the depth estimation one, with the differences that ground truth maps represent normalized vectors, *i.e.* are of size 3 and $L_2$ normalized. The structure of the auxiliary block is therefore close too: we apply the same strategy as for the depth estimation task separately for each component of the vectors (*i.e.* $K_{normal} = 3K_{depth}$ and the channel-wise pooling is applied for each block of $K_{depth}$ maps) and concatenate the resulting three maps. The loss is different however: it is the sum of negative dot product and $L_2$ losses (Dharmasiri et al. 2017), following a $L_2$ normalization layer.

**Fusion of features.** Once intermediate features are extracted from all auxiliary tasks, they are all fused into the base feature maps to yield its refined version. As shown in Figure 4.2 and in Equation 4.1, we do it here with a generic element-wise addition of all feature maps. The optimal fusion scheme could depend on the nature of primary and auxiliary tasks. For example, element-wise product can be interpreted as a gating mechanism (Droniou et al. 2013; Hochreiter et al. 1997), which is well suited when the auxiliary task can be interpreted as an attention map. We show in the experiments that this fusion strategy is relevant for leveraging depth information. Finally, more complex fusion models, *e.g.* full bilinear fusion schemes (Fukui et al. 2016; Ben-Younes et al. 2017), could certainly be leveraged in our context.

## 4.5 Experiments

In this section, we first present an ablation study of ROCK (Section 4.5.1) to evaluate the effect of every component, then we compare ROCK to other state-of-the-art object detection methods on NYUv2 dataset (Section 4.5.2) and using another large-scale dataset (Section 4.5.3). We finally conduct several further experiments to finely analyze ROCK (Section 4.5.4).

**Experimental setup.** We use the indoor dataset NYUv2 (Silberman et al. 2012) for the experiments. This dataset contains relatively few images compared to large-scale datasets, *e.g.* ImageNet (Russakovsky et al. 2015), so additional supervision might yield a larger gain than on bigger datasets. It is composed of an official train/test split with 795 and 654 images respectively. For model analysis and ablation study, we further divide the train set into new train and val sets of 673 and 122 images respectively, taking care that images from a same sequence are

all put into the same set. We then train our model on the train and val sets and evaluate it on the official test split for comparison with state of the art.

Object detection is performed on the same 19 object classes as (Gupta, Girshick, et al. 2014; Hoffman et al. 2016) and is evaluated with three common metrics (mean Average Precision (mAP)@0.5, mAP@0.75 and mAP@[0.5:0.95]) to thoroughly analyze proposed improvements. As additional auxiliary tasks, we use scene classification into $S = 27$ scene classes, depth estimation and surface normal estimation. The ground truths for the first two tasks are provided along with NYUv2 dataset, and we use the targets computed by (Silberman et al. 2012) for the normal prediction task.

We use the SSD framework (W. Liu et al. 2016) with a ResNet-50 (He, X. Zhang, et al. 2016a) backbone architecture pre-trained on ImageNet (Russakovsky et al. 2015). Detection is performed on the output of the conv5 block of ResNet (or its refined version when using our auxiliary block) and on 6 additional feature maps randomly initialized. We train the networks using Adam optimizer (Kingma et al. 2015) with a batch size of 8 for 30,000 iterations with a learning rate of $5 \cdot 10^{-5}$, then we lower it to $5 \cdot 10^{-6}$ and keep training for 10,000 more iterations. We use the data augmentation from SSD (W. Liu et al. 2016) but with fixed aspect ratio for the crops. All examples are then resized to $480 \times 640$ pixels, flipped with probability 0.5, and some color data augmentation (Krizhevsky et al. 2012) is finally applied. Annotations for depth and surface normal estimation are modified accordingly to keep geometries of the scenes (Eigen et al. 2015). Classification and localization losses have weights of 1 and 3 respectively. Loss weights of auxiliary tasks are set to 3 for scene classification and depth estimation, and to 30 for normal estimation (a factor of 10 is advised for this task (Eigen et al. 2015)). Finally, we use the same matching strategy and classification prior as (T.-Y. Lin, Goyal, et al. 2017), and post-process detections with Non-Maximum Suppression (NMS) using a threshold of 0.3.

### 4.5.1 Ablation Study

We present an ablation study of ROCK in Table 4.1 to identify the influence of each component. The first row shows results of our baseline, which is a ResNet SSD model. The last row corresponds to our full ROCK model, which yields improvements of 6.4, 1.3 and 2.3 points in all three metrics with respect to the baseline. To break down this gain between using additional supervision and using our auxiliary block, we first consider a simple flat multi-task SSD baseline, presented in the second row of Table 4.1. The task-specific heads applied on conv5 feature map are just $1 \times 1$ convolution into $S$, 1 or 3 maps depending on the task, followed by a global average pooling for scene classification. This model has an improvement of 3.1, 0.2 and 1.2 with respect to the baseline, which corresponds to the use of additional annotations, *i.e.* the gain from MTL. Then we use our

| Model | | | | Results | | |
|---|---|---|---|---|---|---|
| Name | Auxiliary annotations | Aux. task encoding | Feature merging | mAP@ 0.5 | mAP@ 0.75 | mAP@ [0.5:0.95] |
| Detection baseline | | | | 31.2 | 15.8 | 16.2 |
| Flat MTL baseline | ✓ | | | 34.3 | 16.0 | 17.4 |
| ROCK w/o fusion | ✓ | ✓ | | 35.7 | 16.2 | 17.4 |
| ROCK | ✓ | ✓ | ✓ | **37.6** | **17.1** | **18.5** |

Table 4.1 – **Ablation study of ROCK** on NYUv2 val set in average precision (%).

residual auxiliary block but remove the feature merging step, *i.e.* the decoder and fusion, while keeping the encoder and predictor the same. This is shown in the third row of Table 4.1. This results in an improvement of 1.2, 0.2 for the first two metrics with respect to the flat MTL baseline, which is specifically brought by our auxiliary block, compared to a more common way of doing MTL. The difference between our full ROCK model and this last one, *i.e.* 1.9, 0.9 and 1.1 points on all metrics, is therefore due to the feature merging step, therefore validating the explicit exploitation of auxiliary features through fusion for object detection.

### 4.5.2 Comparison with State of the Art

We compare ROCK to other state-of-the-art object detection methods on NYUv2 dataset in Table 4.2. The first two entries (Gupta, Girshick, et al. 2014; C. Wang et al. 2016) of the table use detection annotations only. It is noticeable that all other methods, leveraging some kind of additional information, outperform them by a large margin, indicating that augmenting images with more annotations has a large impact on this dataset with few examples. Our ROCK model outperforms Modality Hallucination network (Hoffman et al. 2016) by 3.1 points in the same setting, where only depth is used as privileged information. This validates that our approach is able to exploit correlations between depth estimation and object detection. ROCK is also competitive with methods using depth during inference too, *i.e.* not as privileged information (C. Wang et al. 2016), even when they are trained on additional synthetic data (Gupta, Girshick, et al. 2014), as displayed on the following two rows.

Using more annotations yields significantly better results again, as shown with the use of surface normal and curvature (Gupta, Arbeláez, et al. 2015; C. Wang et al. 2016). When ROCK adds supervision from surface normal estimation and scene classification, results are greatly improved, by 2.7 points with respect to using depth only. By specifically designing the architecture to leverage this auxiliary supervision to improve the primary object detection performance, ROCK even

| Model | mAP | btub | bed | bshelf | box | chair | counter | desk | door | dresser | gbin | lamp | monitor | nstand | pillow | sink | sofa | table | tv | toilet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RGB R-CNN (Gupta, Girshick, et al. 2014) | 22.5 | 16.9 | 45.3 | 28.5 | 0.7 | 25.9 | 30.4 | 9.7 | 16.3 | 18.9 | 15.7 | 27.9 | 32.5 | 17.0 | 11.1 | 16.6 | 29.4 | 12.7 | 27.4 | 44.1 |
| RGB R-CNN (C. Wang et al. 2016) | 22.8 | 16.2 | 41.0 | 28.0 | 0.7 | 27.4 | 34.6 | 8.4 | 15.2 | 16.9 | 16.5 | 25.9 | 38.4 | 12.1 | 15.0 | 27.5 | 28.2 | 10.6 | 24.9 | 44.8 |
| Modality Hallucination (D) (Hoffman et al. 2016) | 34.0 | 16.8 | 62.3 | 41.8 | 2.1 | 37.3 | 43.4 | 15.4 | 24.4 | 39.1 | 22.4 | 30.3 | 46.6 | 30.9 | 27.0 | 42.9 | 46.2 | 22.2 | 34.1 | 60.4 |
| ROCK (D) (ours) | 37.1 | 23.5 | 61.8 | **43.0** | 1.5 | 51.8 | 42.5 | 19.5 | **35.7** | 22.9 | 39.0 | 39.8 | 40.0 | 37.7 | 38.5 | 36.6 | 49.8 | 22.0 | 47.1 | 53.1 |
| RGB-D R-CNN (D*) (C. Wang et al. 2016) | 35.5 | 37.8 | 69.9 | 33.9 | 1.5 | 43.2 | 45.0 | 15.7 | 20.5 | 32.9 | 32.9 | 33.7 | 50.9 | 31.6 | 37.3 | 39.0 | 49.0 | 22.9 | 32.2 | 44.9 |
| RGB-D R-CNN (D*+SYN) (Gupta, Girshick, et al. 2014) | 37.3 | 44.4 | 71.0 | 32.9 | 1.4 | 43.3 | 44.0 | 15.1 | 24.5 | 30.4 | 39.4 | 36.5 | 52.6 | 40.0 | 34.8 | 36.1 | 53.9 | 24.4 | 37.5 | 46.8 |
| Pose CNN (DN*+SYN) (Gupta, Arbeláez, et al. 2015) | 38.8 | 36.4 | 70.8 | 35.1 | **3.6** | 47.3 | 46.8 | 14.9 | 23.3 | 38.6 | **43.9** | 37.6 | 52.7 | 40.7 | 42.4 | 43.5 | 51.6 | 22.0 | 38.0 | 47.7 |
| RGB-Geo R-CNN (DNC*) (C. Wang et al. 2016) | 39.3 | 41.8 | 75.0 | 36.4 | 2.2 | 46.9 | 46.4 | 15.8 | 23.9 | 37.9 | 39.9 | 37.5 | 53.0 | 41.7 | 44.0 | 44.4 | 51.8 | 26.9 | 34.5 | 47.0 |
| ROCK (DNS) (ours) | 39.8 | 22.7 | 66.9 | 40.0 | 3.2 | 51.5 | 41.8 | 16.6 | 33.7 | 34.7 | 37.4 | **43.3** | 38.8 | 47.0 | 41.7 | 43.8 | 52.1 | 23.7 | 53.7 | 63.3 |
| ROCK (DNS+MLT) (ours) | **46.8** | **45.8** | **77.4** | 40.8 | 3.2 | **60.2** | **48.4** | **30.1** | **35.7** | **42.6** | 43.1 | 39.7 | **54.3** | **60.4** | **45.4** | **44.9** | **63.0** | **32.5** | **55.0** | **66.2** |

Table 4.2 – **Detailed detection results on NYUv2 test set** in average precision (%) with an IoU threshold of 0.5. Additional supervision used for training is indicated between parenthesis (D: depth, N: surface normals, C: surface curvature, S: scene class). A * means that additional information is also used during inference. Methods marked with (+SYN) and (+MLT) are trained with additional synthetic data and pre-trained on MLT dataset (Y. Zhang et al. 2017) respectively.

outperforms methods using similar kinds of annotations, but at test-time too, in contrast with the privileged context of ROCK.

### 4.5.3 Pre-Training on Large-Scale MLT Dataset

To test ROCK in a more challenging context, we pre-train it on large-scale MLT dataset (Y. Zhang et al. 2017), then fine-tune it and evaluate it on NYUv2 dataset. MLT is composed of over 500,000 synthetic indoor images similar to these from NYUv2, and annotated for object detection, depth and surface normal estimation. This makes this dataset well suited for transfer to NYUv2. However, it raises three main challenges. First, the scale of the dataset is several orders of magnitude larger. In contrast to NYUv2, MLT images are synthetic, so pre-training and transferring models requires to address the domain shift between the two datasets. MLT also does not provide scene classes and ROCK would then have to handle imbalance between pre-trained and newly added tasks when transfered from MLT to NYUv2. We keep the same setup as before but ROCK is learned on 23 slightly different object classes, for 240,000 and 80,000 iterations with the same learning rates. The scene classification branch is removed as there is no annotation for it. Results are presented in the last row of Table 4.2. Transferring from MLT to NYUv2 gives an outstanding state-of-the-art performance of 46.8 points, which is an improvement of 7.0 points over directly training on NYUv2. This result shows that ROCK is able to overcome challenges associated with MLT, in particular to scale to larger datasets and to handle heterogeneous data and missing annotation modalities.

### 4.5.4 Further Analysis

**Complexity of ROCK.** We conduct an analysis of the complexity of ROCK with ResNet-50 as backbone architecture. A comparison of numbers of parameters and inference times without and with ROCK is displayed in Table 4.3. It shows that including ROCK into the network only yields a slight increase in complexity (around 17% more parameters and 7% slower in time), easing its integration into existing models.

| Model | Number of parameters | Inference time (ms/image) |
|---|---|---|
| Detection baseline | 27.8M | 57 |
| ROCK | 32.6M | 61 |

Table 4.3 – **Complexity of ROCK** in parameters and inference time measured with ResNet-50 backbone.

**Analysis of architecture of residual auxiliary block.** We present several design experiments to validate the architecture of our residual auxiliary block in Table 4.4. We first verify that the performance improvement of ROCK is not due to the additional parameters introduced in the model. For this, we evaluate ROCK with the complete architecture but with all auxiliary loss weights set to 0, effectively deactivating MTL. The results are shown on the left of Table 4.4 and are close to those of the detection baseline, indicating that the auxiliary block is only useful to learn from auxiliary tasks in an effective way. We study the effect of the fusion operation with depth only on the right part of Table 4.4. It appears that the product is superior to the addition for this task. Depth bringing a geometric information, the product can be interpreted as a spatial selection. However, design of this component has not been fully explored and further experiments should yield better results.

| Model | mAP@ 0.5 | mAP@ 0.75 | mAP@ [0.5:0.95] |
|---|---|---|---|
| ROCK | 37.6 | 17.1 | 18.5 |
| ROCK w/o aux. sup. | 30.6 | 15.6 | 16.2 |

| Model (depth-only) | mAP@ 0.5 | mAP@ 0.75 | mAP@ [0.5:0.95] |
|---|---|---|---|
| El.-wise addition | 30.9 | 14.8 | 16.1 |
| El.-wise product | 32.3 | 16.2 | 17.3 |

Table 4.4 – **Analysis of architecture of ROCK** on NYUv2 val set in average precision (%).

**Effectiveness of additional supervision.** We here analyze the relation between getting more images or additional annotations on available images. To this end, we train ROCK on a fraction of the train set and observe how many examples are needed to get the same performance as the detection baseline (*i.e.* without auxiliary supervision) on the whole train set. Results are summarized in Table 4.5. Training ROCK on around 70% of the train set roughly gives similar results than the detection baseline (depending on which metric is used for comparison), *i.e.* having the additional three auxiliary tasks to learn from compensates for the loss of 30% of examples. This result shows that fully annotating available data with more tasks can be helpful in domains where examples are hard to obtain.

**Visualization of results.** We show outputs of ROCK on some unseen images in Figure 4.3 for qualitative visual inspection . In the first row, the baseline model wrongly detects a table. However, the classification of the scene into the *bathroom* class might decrease the probability of such an object class, in favor to classes seen more often in these scenes. It is noticeable that detections produced by ROCK agree more with the scene class. On the second row, ROCK detects more objects than the baseline, especially the bed which is only partially visible. This may be due to the depth prediction, where a clear separation of the bed from the rest of the scene is present, easing its detection. In the last row, the pillows are

| Model | mAP@ 0.5 | mAP@ 0.75 | mAP@ [0.5:0.95] |
|---|---|---|---|
| Detection baseline (on 100% of train set) | 31.2 | 15.8 | 16.2 |
| ROCK on 60% of train set | 29.5 | 12.2 | 13.9 |
| ROCK on 70% of train set | 32.8 | 14.5 | 15.9 |
| ROCK on 80% of train set | 34.7 | 16.2 | 17.0 |

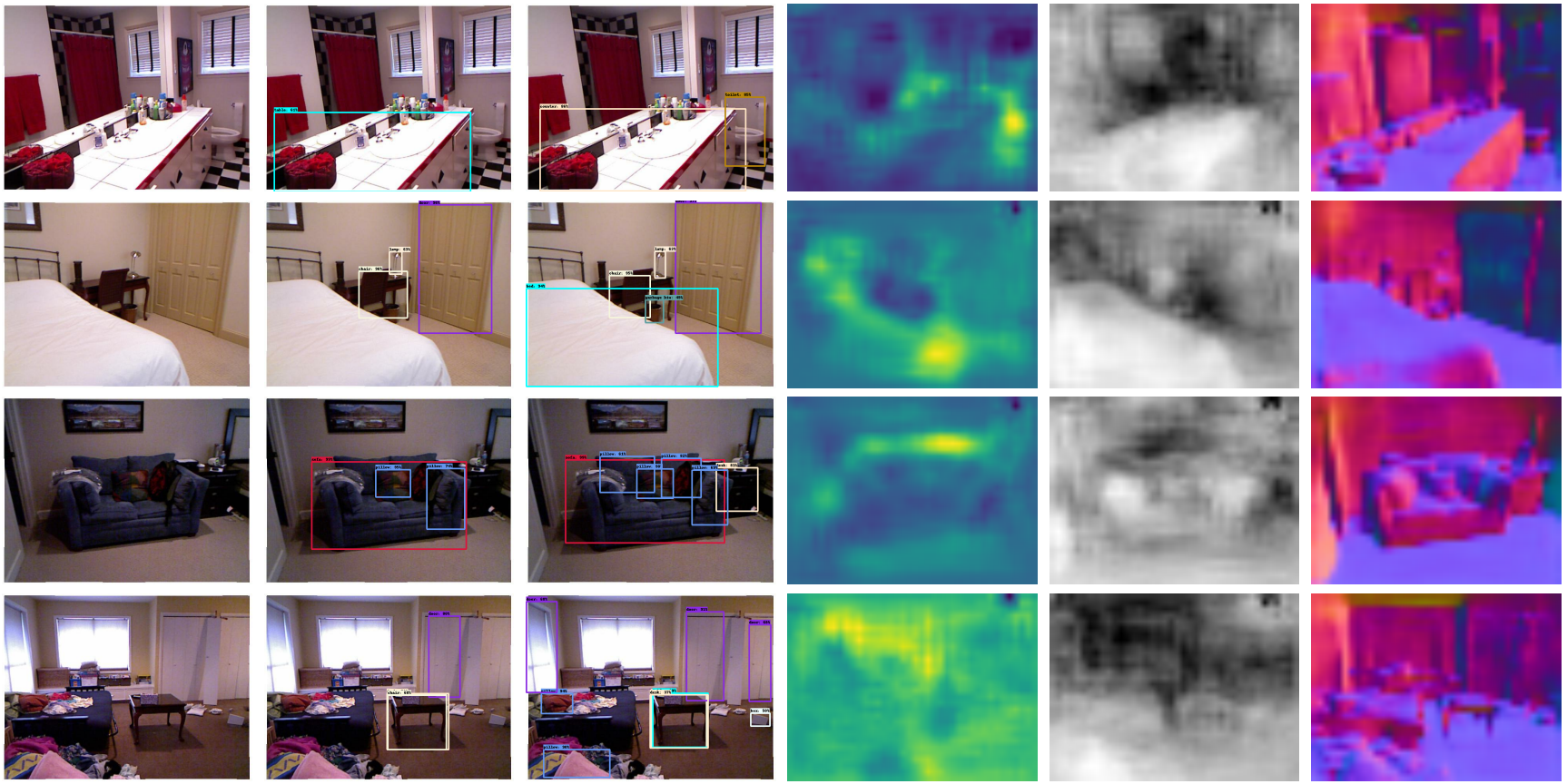
Table 4.5 – **Effectiveness of additional supervision** on NYUv2 val set in average precision (%).

rather difficult to distinguish as they all have similar colors. The surface normal prediction brings geometric information enabling to discern instances and find their contours more easily, leading to better detections. Additional examples are presented in Figure 4.4.

## 4.6 Conclusion

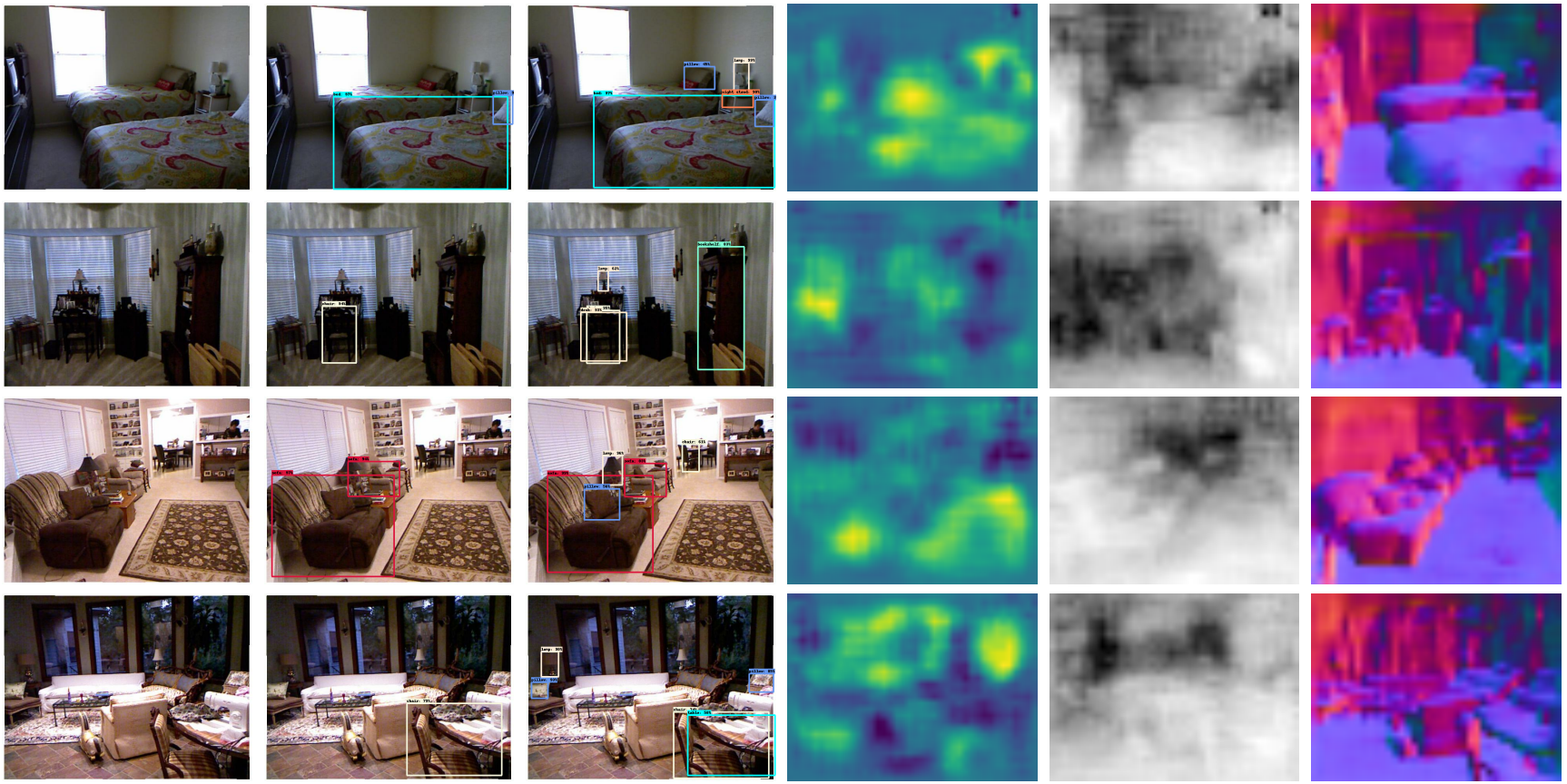
This chapter has introduced ROCK, a generic multi-modal fusion block for deep networks, to tackle the primary MTL context, where auxiliary tasks are leveraged during training to improve performance on a primary task. By designing it with a residual connection and intensive pooling operators in predictors, we maximize the impact and complementarity of the auxiliary representations, benefiting the primary task. We have explored using auxiliary annotations as an alternative to getting more data, which can be useful when there are few training images. When applied to object detection with scene classification, depth and surface normal estimation as auxiliary tasks on NYUv2 dataset, exploiting additional supervision with ROCK yields the same performance than having around 30% additional examples with a single-task model, encouraging to fully exploit available data in contexts where images are difficult to gather.

Although results were encouraging, the design of ROCK has been kept fairly simple to prove the relevance of the approach, and could be further improved. In all experiments, auxiliary supervision has been used during training only, in order to learn a better model. However, in some applications, this additional information might be also accessible after training, when the system is used in real situations. In this case, an interesting direction for future work would be to explore ways to exploit this information at inference time when this is possible, in order to refine predictions with all available data.



(a) image (b) baseline (c) ROCK (d) scene (e) depth (f) normal

Figure 4.3 – **Visualization of outputs.** The original images are presented in (a). Outputs of the detection baseline and ROCK are illustrated in (b) and (c) respectively. Column (d) depicts scene classification through heatmaps of ground truth scene classes (*i.e.* the maps just before global average pooling). Columns (e) and (f) show predictions for depth prediction and surface normal estimation respectively.



(a) image (b) baseline (c) ROCK (d) scene (e) depth (f) normal

Figure 4.4 – **Visualization of outputs.** The original images are presented in (a). Outputs of the detection baseline and ROCK are illustrated in (b) and (c) respectively. Column (d) depicts scene classification through heatmaps of ground truth scene classes (*i.e.* the maps just before global average pooling). Columns (e) and (f) show predictions for depth prediction and surface normal estimation respectively.

CHAPTER 5

# CONCLUSION

## Contents



## 5.1 Summary of Contributions

In this thesis, we have identified several limitations in current Deep Learning (DL) approaches, for which we have proposed solutions. These are organized in three main points detailed below.

**Global pooling function.** Precise annotations are useful to transfer models pre-trained on global image classification task to fine-tune them on spatial tasks. However, they are time-consuming to produce, therefore limiting the size of datasets they form. Weakly Supervised Learning (WSL) framework is an option to perform transfer effectively. It assumes that only partial image-level supervision is available and that latent spatial information needs to be learned to complete the task. When used in deep Convolutional Neural Network (ConvNet), the main point of these methods is a global pooling function used to aggregate features from all regions of images to single image-wise predictions, *i.e.* at the same level than supervision.

In Chapter 2, we have exploited negative evidence and better integrated it into pooling functions, by differentiating its contribution from the standard positive evidence commonly used in WSL. This formulation generalizes various previous models of Multiple-Instance Learning (MIL) and its variants. We have also learned structure by relating features to complementary class modalities in a latent way.

In Chapter 3, the WSL context has been applied to bounding boxes, with supervision at object level. Objects have been modeled with deformable part-based representations inspired by Deformable Part-based Model (DPM), which adapts to objects in a latent way, *i.e.* parts are dynamically optimized without part annotation during training. This representation builds invariance for classification and brings geometric description of shapes of objects for localization, which is especially useful for non-rigid categories. Structure on part positions has further

been introduced by taking interactions of deformations into account, for both classification and localization.

**Efficient architectures for region computation.** We have addressed the problem of efficient end-to-end learning of region-level representations in deep ConvNets. Based on Fully Convolutional Network (FCN) architectures, features can be finely localized to match the precision of the representations used.

In Chapter 2, a FCN has been exploited to learn spatially localized features from image-level supervision only. This approach is well adapted with the region selection process used in the global pooling function, as region-level features are shared within images. Localized representations have then been used for WSL spatial tasks in a straightforward way, here pointwise object localization and semantic segmentation, although having been learned in an image classification setup, *i.e.* with global labels only.

In Chapter 3, FCNs have been used as a solution to issues faced by previous attempts at integrating deformable parts into deep ConvNets. Indeed, original deep architectures such as AlexNet do not easily combine with DPM because of fully connected layers. By generalizing Region-based Fully Convolutional Network (R-FCN) architecture to include deformable part-based representations, we have been able to fully leverage deep features in an end-to-end training procedure.

**Learning with auxiliary supervision.** Exploiting additional supervision can be a solution to fight overfitting, especially in situations where there are few training examples, and these are hard to collect. This is the situation faced at Thales in the context of this thesis, where images are subjected to confidentiality issues, but other kinds of information are available.

In Chapter 4, we have formalized this problem as a special kind of Multi-Task Learning (MTL), named primary MTL. To address it, we have proposed to maximize the influence of auxiliary supervision on results, through the use of residual connections between primary and auxiliary representations. We have also considered not to include learnable parameters in the layers exclusive to auxiliary tasks, so that learning capacity focuses on the primary task. This approach has then been applied to object detection. Compared with Chapter 3, dealing with the same task, we have mainly considered situations with fewer training images, but with additional semantic and geometric auxiliary annotations. Initial results suggest that using auxiliary supervision as an alternative to collecting more data is a promising way to reduce overfitting.

## 5.2 Perspectives for Future Work

At the end of this work, several further directions seem to be worth investigating as future work.

**Semi-supervised learning.** Precise annotations are time-consuming to produce, and therefore limit the sizes of annotated datasets by the number of images that can be labeled, as already noted in Section 1.2.4. This in turn increases the risk of overfitting when learning on these datasets. A possible solution is to use coarse annotations, much faster to obtain. In Chapter 2, we have explored WSL to learn from this kind of annotations. Another possibility is to fully label some images only, but not whole datasets. Semi-Supervised Learning (SSL) (X. Zhu 2006) is a framework to deal with partially unlabeled datasets. Here, annotated data are complemented with other unlabeled examples that can be used for regularization or to learn general distribution of data (Rasmus et al. 2015; Robert et al. 2018). This could be used as an alternative, or in complement of WSL to deal with the lack of precise supervision.

**Learning with heterogeneous data.** As we have seen in Chapter 4, it is essential to leverage all available data to maximize performance. However, they might not all have the same format in real applications, with information of really different natures, noisy or even not always accessible. This then sets the challenge of learning a single model from heterogeneous data, combining all their singularities (Kokkinos 2017; Miech et al. 2018). For instance, if different data do not share the same format, it is needed to adapt them to a common one before processing them with the model, or to use different models, each suited to a particular format, and to combine them after. This kind of approach is also used in other tasks on multiple modalities, *e.g.* when learning text-image embeddings (Ben-Younes et al. 2017).

**Generation of synthetic data.** We have seen in Section 1.1.2 that Thales faces several issues with collecting and annotating images, mainly related to confidentiality reasons. A possible solution to deal with the lack of training annotated examples is the automatic generation of such data (Gaidon et al. 2018). With this process, it would be possible to have unlimited amounts of images, labeled with any level of precision without error (including for any auxiliary task), and specialized to the target tasks. There are mainly two kinds of generation. The first one considers physically-based rendering (Y. Zhang et al. 2017), in order to get realistic images. However, these images are often too clean and do not follow the same distribution as real images. A second approach is to learn the generation process. The popular method for this relies on Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). Even though it is possible to refine

generated images so that they look better (Ashish Shrivastava et al. 2017), it is hard to guarantee that they are realistic enough, and do not exhibit aberrant artifacts. In general, this approach raises questions on transfer from synthetic data to real ones, especially about artifact handling and representativeness of the generated dataset.

# BIBLIOGRAPHY


Andrews, Stuart, Ioannis Tsochantaridis, and Thomas Hofmann (2003). "Support Vector Machines for Multiple-Instance Learning". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 13).

Arandjelovic, Relja, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (2016). "NetVLAD: CNN architecture for weakly supervised place recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 26).

Avila, Sandra, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo Araujo (2012). "Pooling in Image Representation: the Visual Codeword Point of View". In: *Computer Vision and Image Understanding (CVIU)* (cit. on p. 6).

Azizpour, Hossein and Ivan Laptev (2012). "Object Detection Using Strongly-Supervised Deformable Part Models". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 836–849 (cit. on p. 47).

Azizpour, Hossein, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson (2016). "Factors of Transferability for a Generic ConvNet Representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.9, pp. 1790–1802 (cit. on pp. 9, 78).

Bearman, Amy, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei (2016). "What's the Point: Semantic Segmentation with Point Supervision". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 12).

Bell, Sean, Lawrence Zitnick, Kavita Bala, and Ross Girshick (2016). "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 50, 67–69).

Bency, Archith, Heesung Kwon, Hyungtae Lee, S Karthikeyan, and BS Manjunath (2016). "Weakly Supervised Localization using Deep Feature Maps". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 27, 32, 38).

Ben-Younes, Hedi, Rémi Cadène, Nicolas Thome, and Matthieu Cord (2017). "MUTAN: Multimodal Tucker Fusion for Visual Question Answering". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 59, 83, 95).

Bilen, Hakan and Andrea Vedaldi (2016a). "Integrated perception with recurrent multi-task neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 235–243 (cit. on pp. 15, 78).

Bilen, Hakan and Andrea Vedaldi (2016b). "Weakly Supervised Deep Detection Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 27).

Bishop, Christopher (2006). *Pattern Recognition and Machine Learning*. Springer Series in Information Science and Statistics. Berlin, Heidelberg: Springer-Verlag (cit. on p. 5).

Blot, Michael, Matthieu Cord, and Nicolas Thome (2016). "Max-min convolutional neural networks for image classification". In: *IEEE International Conference on Image Processing (ICIP)* (cit. on p. 12).

Boser, Bernhard, Isabelle Guyon, and Vladimir Vapnik (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Workshop on Computational Learning Theory*, pp. 144–152 (cit. on p. 5).

Caruana, Rich (1997). "Multitask Learning". In: *Machine Learning* 28.1, pp. 41–75 (cit. on pp. 14, 15, 18, 76, 78).

Chandra, Siddhartha, Nicolas Usunier, and Iasonas Kokkinos (2017). "Dense and Low-Rank Gaussian CRFs Using Deep Embeddings". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 48).

Chatfield, Ken, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman (2014). "Return of the devil in the details: Delving deep into convolutional nets". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on p. 9).

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille (2015). "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 32, 48, 50).

Chollet, François (2017). "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 9).

Dai, Jifeng, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun (2016). "Instance-Sensitive Fully Convolutional Networks". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 534–549 (cit. on pp. 28, 29, 46, 47).

Dai, Jifeng, Yi Li, Kaiming He, and Jian Sun (2016). "R-FCN: Object Detection via Region-based Fully Convolutional Networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 12, 20, 28–30, 45, 47, 50, 51, 53, 56, 60, 61, 67–70).

Dai, Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei (2017). "Deformable Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 49, 69).

Dalal, Navneet and Bill Triggs (2005). "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 10).

Deselaers, Thomas and Vittorio Ferrari (2010). "A Conditional Random Field for Multiple-Instance Learning". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 17).

Dharmasiri, Thanuja, Andrew Spek, and Tom Drummond (2017). "Joint Prediction of Depths, Normals and Surface Curvature from RGB Images using CNNs".

In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)* (cit. on pp. 78, 83).

Dietterich, Thomas, Richard Lathrop, and Tomás Lozano-Pérez (1997). "Solving the Multiple Instance Problem with Axis-parallel Rectangles". In: *Artificial Intelligence* (cit. on pp. 13, 24).

Droniou, Alain and Olivier Sigaud (2013). "Gated Autoencoders with Tied Input Weights". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 83).

Durand, Thibaut, Taylor Mordan, Nicolas Thome, and Matthieu Cord (2017). "WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 21, 24, 47, 50).

Durand, Thibaut, Nicolas Thome, and Matthieu Cord (2015). "MANTRA: Minimum Maximum Latent Structural SVM for Image Classification and Ranking". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 12, 13, 16, 18, 25, 26, 31).

Durand, Thibaut, Nicolas Thome, and Matthieu Cord (2016). "WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 13, 16, 18, 19, 25, 26, 31, 32, 34–37, 40).

Durand, Thibaut, Nicolas Thome, and Matthieu Cord (2018a). "Exploiting Negative Evidence for Deep Latent Structured Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (cit. on p. 26).

Durand, Thibaut, Nicolas Thome, and Matthieu Cord (2018b). "SyMIL: MinMax Latent SVM for Weakly Labeled Data". In: *IEEE Transactions on Neural Networks and Learning Systems* (cit. on p. 13).

Eigen, David and Rob Fergus (2015). "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2650–2658 (cit. on pp. 78, 84).

Erhan, Dumitru, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov (2014). "Scalable object detection using deep neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 10).

Everingham, Mark, Ali Eslami, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman (2015). "The PASCAL Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision (IJCV)* 111.1, pp. 98–136 (cit. on pp. 33, 38, 45, 60).

Felzenszwalb, Pedro, Ross Girshick, David McAllester, and Deva Ramanan (2010). "Object Detection with Discriminatively Trained Part-based Models". In: *IEEE*

*Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32.9, pp. 1627–1645 (cit. on pp. 13, 16, 29, 45, 46, 51, 53, 65, 70).

Fidler, Sanja, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun (2013). "Bottom-up Segmentation for top-down Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3294–3301 (cit. on p. 47).

Fournier, Jérôme, Matthieu Cord, and Sylvie Philipp-Foliguet (2001). "RETIN: A Content-based Image Indexing and Retrieval System". In: *Pattern Analysis & Applications* 4.2-3, pp. 153–173 (cit. on p. 6).

Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach (2016). "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding". In: *arXiv:1606.01847* (cit. on p. 83).

Fukushima, Kunihiko (1980). "Neocognitron: a Self Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". In: *Biological Cybernetics* 36.4, pp. 193–202 (cit. on pp. 2, 7).

Gaidon, Adrien, Antonio Lopez, and Florent Perronnin (2018). "The Reasonable Effectiveness of Synthetic Visual Data". In: *International Journal of Computer Vision (IJCV)* 126.9, pp. 899–901 (cit. on p. 95).

Gao, Yang, Oscar Beijbom, Ning Zhang, and Trevor Darrell (2016). "Compact Bilinear Pooling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 34, 35).

Gidaris, Spyros and Nikos Komodakis (2015). "Object Detection via a Multi-Region and Semantic Segmentation-Aware CNN Model". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1134–1142 (cit. on pp. 67, 68).

Gidaris, Spyros and Nikos Komodakis (2016a). "Attend Refine Repeat: Active Box Proposal Generation via In-Out Localization". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on p. 60).

Gidaris, Spyros and Nikos Komodakis (2016b). "LocNet: Improving Localization Accuracy for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 60, 68).

Girshick, Ross (2015). "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448 (cit. on pp. 11, 15, 18, 44, 47, 51, 56–60, 67, 68).

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587 (cit. on pp. 11, 12, 18, 44, 57–59).

Girshick, Ross, Forrest Iandola, Trevor Darrell, and Jitendra Malik (2015). "Deformable Part Models are Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 437–446 (cit. on pp. 14, 17, 18, 29, 47, 50).

Gkioxari, Georgia, Ross Girshick, and Jitendra Malik (2015). "Contextual action recognition with R*CNN". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 27).

Goh, Hanlin, Nicolas Thome, Matthieu Cord, and Joo-Hwee Lim (2014). "Learning Deep Hierarchical Visual Feature Coding". In: *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (cit. on p. 26).

Gong, Yunchao, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik (2014). "Multi-scale Orderless Pooling of Deep Convolutional Activation Features". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 18, 26, 34, 35).

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 95).

Guerry, Joris, Bertrand Le Saux, and David Filliat (2017). ""Look At This One" Detection sharing between modality-independent classifiers for robotic discovery of people". In: *Proceedings of the European Conference on Mobile Robotics (ECMR)*, pp. 1–6 (cit. on p. 79).

Gupta, Saurabh, Pablo Arbeláez, Ross Girshick, and Jitendra Malik (2015). "Inferring 3D object pose in RGB-D images". In: *arXiv:1502.04652* (cit. on pp. 79, 85, 86).

Gupta, Saurabh, Ross Girshick, Pablo Arbeláez, and Jitendra Malik (2014). "Learning rich features from RGB-D images for object detection and segmentation". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 345–360 (cit. on pp. 79, 84–86).

Hane, Christian, Lubor Ladicky, and Marc Pollefeys (2015). "Direction matters: Depth estimation with a surface normal classifier". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 381–389 (cit. on p. 78).

Hariharan, Bharath, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik (2011). "Semantic Contours from Inverse Detectors". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 39).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc. (cit. on p. 5).

Hazirbas, Caner, Lingni Ma, Csaba Domokos, and Daniel Cremers (2016). "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture". In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 213–228 (cit. on p. 79).

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 12).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 37.9, pp. 1904–1916 (cit. on pp. 18, 26, 69).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016a). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 7–9, 26, 28, 29, 31–35, 40, 50, 59, 65, 67, 84).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016b). "Identity Mappings in Deep Residual Networks". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 9).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural Computation* 9.8, pp. 1735–1780 (cit. on p. 83).

Hoffman, Judy, Saurabh Gupta, and Trevor Darrell (2016). "Learning with Side Information through Modality Hallucination". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 826–834 (cit. on pp. 19, 78, 84–86).

Hornik, Kurt (1991). "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural networks* 4.2, pp. 251–257 (cit. on p. 7).

Howard, Andrew, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *arXiv:1704.04861* (cit. on p. 9).

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Weinberger (2017). "Densely Connected Convolutional Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 9).

Iandola, Forrest, Song Han, Matthew Moskewicz, Khalid Ashraf, William Dally, and Kurt Keutzer (2016). "SqueezeNet: AlexNet-Level Accuracy with 50x fewer Parameters and< 0.5 mb Model Size". In: *arXiv:1602.07360* (cit. on p. 9).

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, et al. (2015). "Spatial Transformer Networks". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 12, 26).

Jégou, Hervé, Matthijs Douze, Cordelia Schmid, and Patrick Pérez (2010). "Aggregating Local Descriptors into a Compact Image Representation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 6).

Kingma, Diederik and Jimmy Ba (2015). "Adam: A method for stochastic optimization". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 84).

Kokkinos, Iasonas (2017). "UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 15, 18, 78, 95).

Kolesnikov, Alexander and Christoph Lampert (2016). "Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 39).

Kong, Tao, Anbang Yao, Yurong Chen, and Fuchun Sun (2016). "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 50, 68).

Krähenbühl, Philipp and Vladlen Koltun (2011). "Efficient Inference in Fully Connected CRFs with Gaussian Ddge Potentials". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 109–117 (cit. on pp. 39, 48, 54, 56).

Krause, Jonathan, Timnit Gebru, Jia Deng, Li-Jia Li, and Fei-Fei Li (2014). "Learning Features and Parts for Fine-Grained Recognition." In: *Proceedings of the International Conference on Pattern Recognition (ICPR)* (cit. on p. 27).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105 (cit. on pp. 3, 6–9, 18, 24, 44, 47, 50, 67, 84).

Kulkarni, Praveen, Frédéric Jurie, Joaquin Zepeda, Patrick Pérez, and Louis Chevallier (2016). "SPLeaP: Soft Pooling of Learned Parts for Image Classification". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 12, 34, 35).

Ladicky, Lubor, Jianbo Shi, and Marc Pollefeys (2014). "Pulling things out of perspective". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 89–96 (cit. on p. 78).

Lafferty, John, Andrew McCallum, and Fernando Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 54).

Lai, Kuan-Ting, Felix Yu, Ming-Syan Chen, and Shih-Fu Chang (2014). "Video Event Detection by Inferring Temporal Instance Labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 26).

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper Depth Prediction with Fully Convolutional Residual Networks". In: *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pp. 239–248 (cit. on p. 83).

Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce (2006). "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 6, 33).

LeCun, Yann, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural computation* 1.4, pp. 541–551 (cit. on pp. 2, 7, 44).

Li, Li-Jia, Hao Su, Eric Xing, and Li Fei-Fei (2010). "Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 32).

Li, Weixin and Nuno Vasconcelos (2015). "Multiple Instance Learning for Soft Bags via Top Instances". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 13, 16, 25, 26, 31).

Li, Yi, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei (2017). "Fully Convolutional Instance-aware Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 28, 29, 46, 47, 50).

Lin, Di, Xiaoyong Shen, Cewu Lu, and Jiaya Jia (2015). "Deep LAC: Deep Localization, Alignment and Classification for Fine-Grained Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1666–1674 (cit. on pp. 27, 46).

Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2017). "Feature Pyramid Networks for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 69).

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017). "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 12, 69, 84).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 740–755 (cit. on pp. 33, 38, 69).

Liu, Beyang, Stephen Gould, and Daphne Koller (2010). "Single image depth estimation from predicted semantic labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1253–1260 (cit. on p. 78).

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed (2016). "SSD: Single Shot MultiBox Detector". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 11, 12, 68, 77, 84).

Lloyd, Stuart (1982). "Least Squares Quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137 (cit. on p. 6).

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440 (cit. on pp. 20, 27, 47, 50).

Lopez-Paz, David, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik (2016). "Unifying Distillation and Privileged Information". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 78).

Lowe, David (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision (IJCV)* 60.2, pp. 91–110 (cit. on p. 6).

Lu, Yongxi, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Feris (2017). "Fully-Adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1131–1140 (cit. on p. 15).

Ma, Wei-Ying and Bangalore Manjunath (1999). "Netra: A Toolbox for Navigating Large Image Databases". In: *Multimedia Systems* 7.3, pp. 184–198 (cit. on p. 6).

McCulloch, Warren and Walter Pitts (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The Bulletin of Mathematical Biophysics* 5.4, pp. 115–133 (cit. on p. 7).

Meyerson, Elliot and Risto Miikkulainen (2018). "Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 15, 78).

Miech, Antoine, Ivan Laptev, and Josef Sivic (2018). "Learning a Text-Video Embedding from Imcomplete and Heterogeneous Data". In: *arXiv:1804.02516* (cit. on p. 95).

Misra, Ishan, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert (2016). "Cross-Stitch Networks for Multi-task Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3994–4003 (cit. on p. 15).

Mordan, Taylor, Nicolas Thome, Matthieu Cord, and Gilles Henaff (2017). "Deformable Part-based Fully Convolutional Network for Object Detection". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on pp. 21, 44).

Mordan, Taylor, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018a). "End-to-End Learning of Latent Deformable Part-Based Representations for Object Detection". In: *International Journal of Computer Vision (IJCV)* (cit. on pp. 21, 44).

Mordan, Taylor, Nicolas Thome, Gilles Henaff, and Matthieu Cord (2018b). "Revisiting Multi-Task Learning with ROCK: a Deep Residual Auxiliary Block for Visual Detection". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 21, 76).

Oquab, Maxime, Léon Bottou, Ivan Laptev, and Josef Sivic (2014). "Learning and Transferring Mid-Level Image Representations using Convolutional Neural

Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 18).

Oquab, Maxime, Léon Bottou, Ivan Laptev, and Josef Sivic (2015). "Is Object Localization for Free? – Weakly-Supervised Learning with Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 13, 16, 18, 24, 26, 31, 32, 34, 35, 38).

Ott, Patrick and Mark Everingham (2011). "Shared parts for deformable part-based models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1513–1520 (cit. on p. 47).

Papandreou, George, Liang-Chieh Chen, Kevin Murphy, and Alan Yuille (2015). "Weakly-and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 27, 39).

Parizi, Sobhan, Andrea Vedaldi, Andrew Zisserman, and Pedro Felzenszwalb (2015). "Automatic Discovery and Optimization of Parts for Image Classification". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 16, 24, 26, 31, 34, 35).

Park, Seong-Jin, Ki-Sang Hong, and Seungyong Lee (2017). "RDFNet: RGB-D Multi-Level Residual Feature Fusion for Indoor Semantic Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 79).

Pathak, Deepak, Philipp Krahenbuhl, and Trevor Darrell (2015). "Constrained Convolutional Neural Networks for Weakly Supervised Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 27, 32, 39).

Pathak, Deepak, Evan Shelhamer, Jonathan Long, and Trevor Darrell (2015). "Fully Convolutional Multi-Class Multiple Instance Learning". In: *Proceedings of the International Conference on Learning Representations (ICLR) Workshop* (cit. on pp. 27, 39).

Pechyony, Dmitry and Vladimir Vapnik (2010). "On the Theory of Learnining with Privileged Information". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1894–1902 (cit. on p. 78).

Perronnin, Florent and Christopher Dance (2007). "Fisher Kernels on Visual Vocabularies for Image Categorization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 6).

Perronnin, Florent, Jorge Sánchez, and Thomas Mensink (2010). "Improving the Fisher Kernel for Large-Scale Image Classification". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 6).

Picard, David and Philippe-Henri Gosselin (2011). "Improving Image Similarity with Vectors of Locally Aggregated Tensors". In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (cit. on p. 6).

Pinheiro, Pedro and Ronan Collobert (2015). "From Image-level to Pixel-level Labeling with Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 27, 39).

Quattoni, Ariadna and Antonio Torralba (2009). "Recognizing Indoor Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 33).

Rasmus, Antti, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko (2015). "Semi-Supervised Learning with Ladder Networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3546–3554 (cit. on p. 95).

Razavian, Ali Sharif, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson (2014). "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop* (cit. on p. 9).

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 12).

Redmon, Joseph and Ali Farhadi (2017). "YOLO9000: Better, Faster, Stronger". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 12).

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99 (cit. on pp. 12, 15, 18, 44, 68).

Ren, Zhongzheng and Yong Jae Lee (2018). "Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 79).

Robert, Thomas, Nicolas Thome, and Matthieu Cord (2018). "HybridNet: Classification and Reconstruction Cooperation for Semi-Supervised Learning". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 95).

Rosenblatt, Frank (1958). "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain". In: *Psychological Review* 65.6, p. 386 (cit. on p. 7).

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252 (cit. on pp. 3, 6, 9, 10, 24, 28, 32, 44, 60, 83, 84).

Salton, Gerard and Michael McGill (1986). "Introduction to Modern Information Retrieval". In: (cit. on p. 6).

Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen (2018). "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 9).

Savalle, Pierre-André, Stavros Tsogkas, George Papandreou, and Iasonas Kokkinos (2014). "Deformable Part Models with CNN Features". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV), Parts and Attributes Workshop* (cit. on pp. 47, 50).

Schapire, Robert (1990). "The Strength of Weak Learnability". In: *Machine Learning* 5.2, pp. 197–227 (cit. on p. 10).

Sermanet, Pierre, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun (2014). "OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 11).

Shang, Wenling, Kihyuk Sohn, Diogo Almeida, and Honglak Lee (2016). "Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 12).

Sharmanska, Viktoriia, Novi Quadrianto, and Christoph Lampert (2013). "Learning to Rank Using Privileged Information". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 78).

Sharmanska, Viktoriia, Novi Quadrianto, and Christoph Lampert (2014). "Learning to Transfer Privileged Information". In: *arXiv:1410.0389* (cit. on p. 78).

Shi, Zhiyuan and Tae-Kyun Kim (2017). "Learning and refining of privileged information-based RNNs for action recognition from depth sequences". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 19, 78).

Shrivastava, Abhinav, Abhinav Gupta, and Ross Girshick (2016). "Training Region-based Object Detectors with Online Hard Example Mining". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 10, 68, 69).

Shrivastava, Ashish, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb (2017). "Learning from Simulated and Unsupervised Images through Adversarial Training." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 96).

Sicre, Ronan, Yannis Avrithis, Ewa Kijak, and Frédéric Jurie (2017). "Unsupervised part learning for visual recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 46).

Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus (2012). "Indoor Segmentation and Support Inference from RGBD Images". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 83, 84).

Simon, Marcel and Erik Rodner (2015). "Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1143–1151 (cit. on p. 46).

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 7–9, 18, 34, 50).

Sivic, Josef and Andrew Zisserman (2003). "Video Google: A Text Retrieval Approach to Object Matching in Videos". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on pp. 6, 26).

Spinello, Luciano and Kai Arras (2012). "Leveraging RGB-D data: Adaptive fusion and domain adaptation for object detection". In: *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pp. 4469–4474 (cit. on p. 79).

Sun, Chen, Manohar Paluri, Ronan Collobert, Ram Nevatia, and Lubomir Bourdev (2016). "ProNet: Learning to Propose Object-Specific Boxes for Cascaded Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 16, 26, 27, 32, 34, 35, 38).

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi (2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (cit. on p. 9).

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going Deeper with Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 7–9).

Szegedy, Christian, Alexander Toshev, and Dumitru Erhan (2013). "Deep Neural Networks for Object Detection". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2553–2561 (cit. on p. 10).

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna (2016). "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 9).

Tucker, Ledyard (1966). "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3, pp. 279–311 (cit. on p. 58).

Uijlings, Jasper, Koen van de Sande, Theo Gevers, and Arnold Smeulders (2013). "Selective Search for Object Recognition". In: *International Journal of Computer Vision (IJCV)* 104.2, pp. 154–171 (cit. on p. 11).

Vapnik, Vladimir (1992). "Principles of Risk Minimization for Learning Theory". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on p. 4).

Vapnik, Vladimir (1995). *The Nature of Statistical Learning Theory*. Springer (cit. on p. 6).

Vapnik, Vladimir and Rauf Izmailov (2015). "Learning Using Privileged Information: Similarity Control and Knowledge Transfer". In: *Journal of Machine Learning Research (JMLR)* 16, pp. 2023–2049 (cit. on p. 78).

Vapnik, Vladimir and Akshay Vashist (2009). "A New Learning Paradigm: Learning Using Privileged Information". In: *Neural Networks* 22.5-6, pp. 544–557 (cit. on pp. 18, 76, 78).

Viola, Paul and Michael Jones (2001). "Rapid Object Detection using a Boosted Cascade of Simple Features". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 10).

Wan, Li, David Eigen, and Rob Fergus (2015). "End-to-End Integration of a Convolution Network, Deformable Parts Model and Non-Maximum Suppression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 851–859 (cit. on pp. 14, 17, 18, 47, 50, 51).

Wang, Chu and Kaleem Siddiqi (2016). "Differential geometry boosts convolutional neural networks for object detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 51–58 (cit. on pp. 79, 85, 86).

Wang, Jinghua, Zhenhua Wang, Dacheng Tao, Simon See, and Gang Wang (2016). "Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 664–679 (cit. on p. 79).

Wang, Peng, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille (2015a). "Joint Object and Part Segmentation Using Deep Learned Potentials". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1573–1581 (cit. on p. 46).

Wang, Peng, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille (2015b). "Towards unified depth and semantic prediction from a single image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2800–2809 (cit. on p. 78).

Wei, Zijun and Minh Hoai (2016). "Region Ranking SVM for Image Classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 27, 34).

Weng, Juyang, Narendra Ahuja, and Thomas Huang (1992). "Cresceptron: a self-organizing neural network which grows adaptively". In: *International Joint Conference on Neural Networks (IJCNN)* (cit. on p. 12).

Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He (2017). "Aggregated Residual Transformations for Deep Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 9, 50, 65, 67).

Xu, Huijuan and Kate Saenko (2016). "Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering". In: *Pro-*

*ceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 26).

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 26).

Yang, Yongxin and Timothy Hospedales (2017). "Deep Multi-task Representation Learning: A Tensor Factorisation Approach". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 15).

Yu, Felix, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang (2013). "∝SVM for Learning with Label Proportions". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on pp. 26, 31).

Yu, Fisher and Vladlen Koltun (2016). "Multi-Scale Context Aggregation by Dilated Convolutions". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on p. 50).

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on pp. 50, 65, 67).

Zagoruyko, Sergey, Adam Lerer, Tsung-Yi Lin, Pedro Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollar (2016). "A MultiPath Network for Object Detection". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on pp. 50, 67, 69).

Zamir, Amir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese (2018). "Taskonomy: Disentangling Task Transfer Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 14, 15, 78).

Zeiler, Matthew and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 818–833 (cit. on p. 9).

Zhang, Han, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas (2016). "SPDA-CNN: Unifying Semantic Part Detection and Abstraction for Fine-Grained Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1143–1152 (cit. on pp. 27, 46).

Zhang, Jianming, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff (2016). "Top-Down Neural Attention by Excitation Backprop". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 26).

Zhang, Ning, Jeff Donahue, Ross Girshick, and Trevor Darrell (2014). "Part-based R-CNNs for Fine-Grained Category Detection". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 834–849 (cit. on pp. 12, 27, 46).

Zhang, Ning, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev (2014). "PANDA: Pose Aligned Networks for Deep Attribute Modeling". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 12).

Zhang, Yinda, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser (2017). "Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 86, 87, 95).

Zheng, Shuai, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr (2015). "Conditional Random Fields as Recurrent Neural Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1529–1537 (cit. on p. 48).

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba (2016). "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 12, 13, 16, 18, 24, 26, 30, 31, 34–36).

Zhou, Bolei, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva (2014). "Learning Deep Features for Scene Recognition using Places Database". In: *Advances in Neural Information Processing Systems (NIPS)* (cit. on pp. 34, 35).

Zhou, Xi, Kai Yu, Tong Zhang, and Thomas Huang (2010). "Image Classification Using Super-Vector Coding of Local Image Descriptors". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on p. 6).

Zhou, Zhi-Hua, Yu-Yin Sun, and Yu-Feng Li (2009). "Multi-Instance Learning by Treating Instances as Non-I.I.D. Samples". In: *Proceedings of the International Conference on Machine Learning (ICML)* (cit. on p. 17).

Zhu, Long, Yuanhao Chen, Alan Yuille, and William Freeman (2010). "Latent Hierarchical Structural Learning for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1062–1069 (cit. on pp. 47, 51).

Zhu, Xiaojin (2006). "Semi-Supervised Learning Literature Survey". In: *Computer Science, University of Wisconsin-Madison* 2.3, p. 4 (cit. on p. 95).

Zitnick, Lawrence and Piotr Dollár (2014). "Edge Boxes: Locating Object Proposals from Edges". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pp. 391–405 (cit. on p. 11).