Ivan Krešo, Petra Bevandić, Marin Oršić, Siniša Šegvić

Convolutional Models for Segmentation and Localization

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia

Abstract

The revival of deep models has profoundly improved the accuracy of image classification models and provided a large improvement potential in related computer vision tasks. Recently, much attention has been directed towards dense prediction models which produce distinct output in each image pixel. This paper addresses two particular instances of dense prediction: object localization and semantic segmentation. We briefly review the underlying operation principles, present some of our experimental results and discuss ways to analyze the success of learning and the utility of the resulting models.

1. Introduction

Recent revival of deep learning has enabled construction of multi-stage computer vision algorithms in which all stages can be trained end-to-end. Most success has been achieved with convolutional models [14] which ensure translational invariance as an essential property of vision. The resulting development has led to artificial vision systems which outperform humans in large-scale image classification [23]. This progress has been steadily followed by advances in other vision tasks. Thus, it has been noticed that semantic segmentation can be carried out by applying the same ImageNet pre-trained classification model in each pixel (cf. Fig.1). The implied computational complexity has been reduced by applying the model layerwise (as opposed to patchwise), in a convolutional manner [24].

However, it turns out that straight-forward convolutional application of a classification model results in a significant reduction of the output resolution. Consequently, a smooth transition from classification to dense prediction is hampered by strict memory limitations of contemporary GPUs as we shall show in the following sections.

2. Semantic segmentation

Semantic segmentation is a computer vision task in which we classify each image pixel into the corresponding high-level class. The ground-truth class labels are determined by the kind of the object or surface which gets projected onto the corresponding pixel. Due to being complementary to object localization, semantic segmentation represents an important step towards advanced future techniques for natural image understanding. Some attractive application fields include autonomous control, intelligent transportation systems and automated analysis of photographs and video.

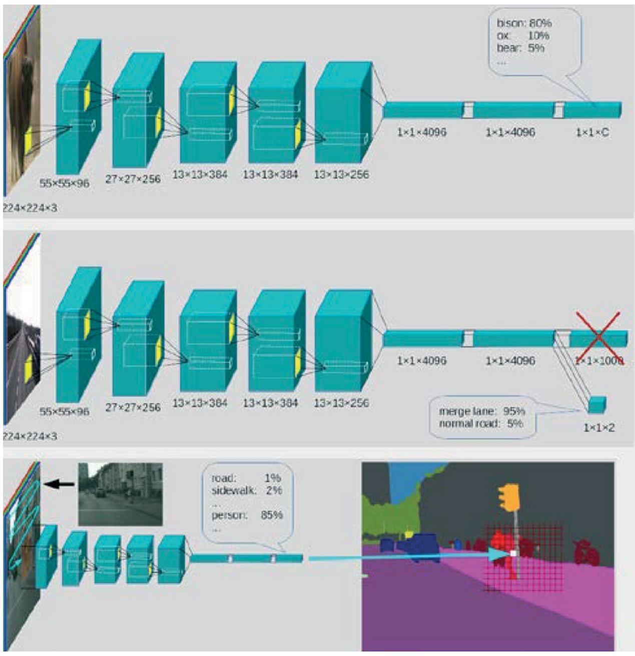

Fig. 1. A convolutional model is usually pre-trained on the ImageNet dataset which comprises 10^6 images and 10^3 classes (top). The model can be easily adapted to a simpler task by fine-tuning on the target dataset (middle). The simplest approach to achieve dense prediction would be to slide the model over all image positions (bottom). In practice we optimize this idea by applying the model *layerwise* in a fully convolutional fashion [24].

2.1. Architectural considerations

When designing an architecture for semantic segmentation we usually start with a network created for image classification. This allows to pretrain the model on the ImageNet dataset, which typically leeds to best results. In image classification task, the model output is a vector representing the distribution over classes for the whole image. In order to repurpose any classification architecture for segmentation, we need to remove the global pooling at the end and replace all fully connected layers with convolutions. However, due to intermediate pooling layers, we still get a 32x subsampled prediction. There are two ways to restore the lost resolution and get predictions at the pixel level. One way is to use intermediate feature maps before pooling layers in each block to recover lost information and refine boundaries in downsampled representation [26]. The other way is to remove pooling layers and introduce dilated convolutions [3] that will preserve the same size of the receptive field. The main downside of the dilated convolution approach is its inefficiency due to large resolution of the deep layers with many feature maps. Another downside is that we are forcing the model to propagate even very small objects through all layers of the network, which leads to potentially losing some model capacity. The upside is its simplicity because it is the easiest way to convert the network from classification to segmentation task. However, we can avoid the problems with dilated convolution and still achieve high prediction density by using ladder-style blending [26] which leverages intermediate feature maps to restore the lost details. We have successfully used this technique to successfully convert the 32x subsampled representation to the 4x subsampled output [13]. Our upsampling subnet is very efficient and introduces only a negligible increase in running time. This is achieved by blending two feature tensors from different subsampling levels with a single 3x3 convolution. It turns out that the pretrained classification network can very well adapt to this simple blending technique during fine-tuning.

2.2 Experiments on the Cityscapes dataset

We evaluated our models on the Cityscapes dataset [5] which consists of 5000 images with fine annotations and 20000 images with coarse annotations. In our experiments we used only the fine annotations. The dataset is labeled with 19 classes. The resolution of the images is 1024x2048 (cf. Fig. 2).

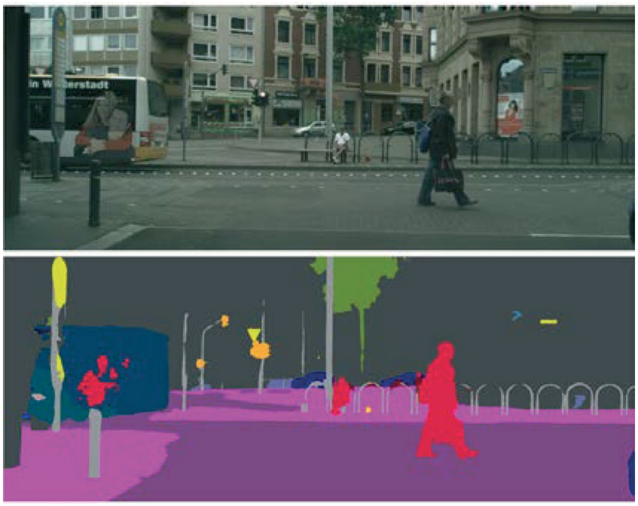

Fig. 2. Original image from Cityscapes test (top) and the dense predictions (bottom) of our semantic segmentation model (purple: road, dark blue: bus, person: red, etc). Note that a commercial sticker on the bus has been erroneously segmented as class person. Although depicting persons, that particular region should be segmented as class bus to which it semantically belongs.

The quality of semantic segmentation models is usually evaluated with intersection over union (IoU). For each class we consider pixels corresponding to the predictions and the ground-truth annotation. The IoU metric is then defined as the ratio between intersection and union of those two areas. Finally, we take the mean IoU across all classes or mIoU for short [8]. Table 1. shows our results on Cityscapes validation subset with models based on the DenseNet-121 architecture [10] pretrained on ImageNet. DenseNet 32x is the baseline model where a 32x subsampled prediction is produced right after the last DenseNet block. LadderDenseNet 4x uses ladder-style feature blending [13]. In Dilated 8x DenseNet 4x we used dilated convolutions in the last two blocks to obtain 8x subsampled prediction followed by one level of ladder-style feature upsampling to produce 4x subsampled output. Note that we couldn't use dilated convolutions to directly obtain 4x prediction due to memory limitations. LadderDenseNet 4x outperforms Dilated 8x DenseNet 4x despite requiring less memory and leading to faster execution. The large improvement between DenseNet 32x and LadderDenseNet 4x reveals the importance of prediction density. We came to similar conclusions in experiments on PASCAL VOC 2012 and CamVid datasets

3. Object localization

The purpose of object localization is to find objects of various classes in the input image and describe them with bounding boxes and class labels. This task is challenging as objects may vary in size, shape, pose, occlusion etc. Existing approaches fall into two groups. Two-stage approaches first perform class-agnostic localization of object candidates. In the second stage, the candidates are classified one at a time. On the other hand, single stage approaches produce dense predictions of bounding boxes and class labels in the compound processing step. Two stage approaches still achieve better accuracy, however we prefer single stage approaches due to simpler design and better execution speed.

3.1. Single shot detector

Single Shot Detector (SSD) [16] is the first one-stage approach to achieve accuracy comparable to two-stage approaches. It enables real time execution on 512x512 images. SSD handles the problem of varying object size by making predictions at suitable layers of a deep image representation. The features are extracted by a convolutional model consisting of the first 5 convolutional blocks from the VGG architecture [25] and 4 additional convolutional blocks. Each block subsamples the resolution of the previous block by the factor of 2. The last 6 levels of representation are connected to multibox heads which perform dense prediction of object classes and bounding box positions. Bounding box predictions are performed for multiple aspect ratios: {0.3, 0.5, 1, 2, 3}.

3.2. Experiments on MOT 2015 dataset

We evaluated the SSD approach on MOT 2015 dataset [15]. We split each training sequence into train and validation subsets such that the last 20% of images in each video are moved to the validation subset. This produces 4334 training images and 1087 validation images (we omit images that do not have any ground truth detections). The training procedure was the same as for SSD300 [16]. We experimented by adding a prediction with a taller aspect ratio (due to the fact that pedestrians are usually in a standing pose) but that did not result in any significant improvement. We display the results in Table 1. We notice a large improvement when training SSD on MOT 2015 train rather than training on 20 object classes from PASCAL VOC 2007 + 2012. Note that the competing algorithms were not tuned on MOT2015: the presented improvement is due to opportunity to better fit our model to the data. This emphasizes the importance of learning on training data whose distribution matches the distribution of the test data.

Sample detections are shown in Figure 3. SSD achieves very good accuracy on large to medium sized object while occasionally having trouble with small or distant objects. The method also has troubles with predicting false positives as well as classifying an object to a wrong but similar class (eg. mistaking a sheep for a cow). Our current experiments show that such problems can be significantly diminished with improved models. However, this research is still incomplete and so we will have to present it elsewhere.

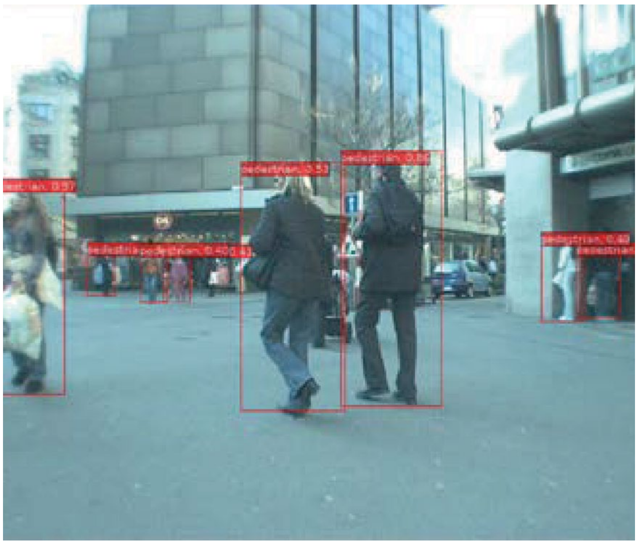

Fig. 3. Pedestrian detections on MOT 2015 val obtained by an SSD model trained on MOT 2015 train. Note that only the smallest three pedestrians have been missed.

4. Analysis of the learned models

Deep models achieve state-of-the-art performance in many computer vision tasks. However our understanding on how and why those models work remains limited. Answering these questions would not only help us improve on existing models (e.g. by understanding why deep models make mistakes), but also could play an important role in real-world application of deep models (e.g. anticipate legal implications of using deep models in practice).

4.1 Feature visualization

One way to answer how deep models work in a human friendly way is by using qualitative representations of different layers in a network. A simple example of qualitative analysis is visualization of filters in the first layer of convolutional networks. This approach is however not useful for units in deeper layers. A simple solution introduced in [erhan09icmlw] is to look for input patterns that maximize the activation of a hidden unit rather than visualizing unit content directly. Defined this way, a feature h can be visualized by locating an image patch $\mathbf{x}^*$ which maximizes its value given the model parameters ϕ:

$$\mathbf{x}^* = \text{argmax}\ \phi. \qquad (1)$$

However, this definition opens up a new set of challenges [olah17distill]. For example, how to choose a hidden unit? Is it more useful to do visualize a single neuron, a single feature map, or the whole layer? Is there more than one pattern that could represent what makes a unit fire (e.g. should a neuron responsible for detecting birds fire for both penguins and hummingbirds)? Furthermore, optimizing just to make units fire does not necessarily lead to interpretable visualizations. This method can also be used to generate examples that the network classifies into one of the possible classes with a high level of confidence, without the input necessarily making visual sense to a human.

We can solve the problem of finding input patterns that maximize the activation of the hidden unit using gradient descent. We usually start from a randomly sampled image, calculate gradient of the output of the hidden unit of interest with respect to the input, and finally apply the gradient to the input. However, basic gradient descent usually gives us uninterpretable images. This problem can be solved by expanding the original problem with a suitable regularizer. Results can be further improved by slightly perturbing the input between optimizations steps to make the final visualization more robust to image transformations. The most common perturbations are blurring, jittering, scaling and rotating the input before calculating the gradient. Fig. 4 shows different types of visualization for a DenseNet architecture fine-tuned for classification on VOC 2007.

Table 1. Semantic segmentation experiments on Cityscapes val. Sx denotes S times subsampled predictions which are subsequently upsampled with bilinear interpolation. All training and evaluation images in this experiment were resized to 1024×448, while the batch size was set to 4.

Model	mIoU (%)
DenseNet 32x	62.52
Dilated 8x DenseNet 4x	71.56
LadderDenseNet 4x	72.82

Table 2. Object localization experiments on MOT 2015 val.

Model	Average Precision
SSD, ImageNet + Pascal0712 [liu16eccv]	57.06%
Agg. channel features, INRIA [dollar14pami]	60.2%
SSD, ImageNet + MOT 2015(ours)	75.5%

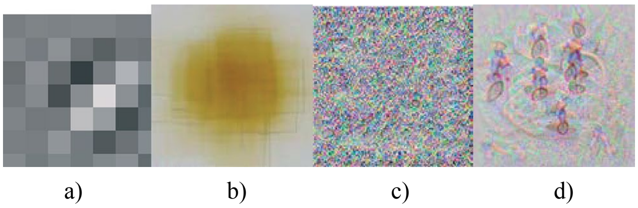

Fig. 4. Different visualizations for DenseNet 121 fine-tuned on Pascal VOC 2007: filter #20 of the first convolution layer (a), input pattern that maximizes its activation (b), input patterns that maximize the class 'bicycle' (c,d). No regularization was used for generating (c), while for (d) we used jittering, scaling, rotating and blurring.

4.2 Adversarial examples

Adversarial examples arise when a given image is purposively modified in a way to disrupt the correct prediction by the target model [2, 1]. If the model is not trained in a defensive manner, imperceptible perturbations can be crafted which cause the model to change its prediction away from the correct class y, while still reporting a high level of confidence. Suppose the model f provides a correct prediction in image $\mathbf{x}_i$: $f(\mathbf{x}_i) = y_i$. Then, we can recover the adversarial perturbation $\mathbf{r}$ by optimizing the following problem:

$$\min \| r \|_2 \; s.t. \; f(\mathbf{x}_i + \mathbf{r}) \neq y_i, \quad \mathbf{x}_i + \mathbf{r} \in [0,1]^m.$$

This problem can be solved by propagating the adversarial gradients to the input image $\mathbf{x}_i$ and subsequently optimizing $\mathbf{x}_i$ with gradient descent. Adversarial images can be crafted for dense prediction models as well, as shown in Figure 5.

Following the discovery of adversarial examples, a number of exploits have been devised in literature, which, in theory, could seriously compromise practical computer vision applications. For instance, an attacker could wreak havoc in autonomous traffic by decorating stop signs with adversarial stickers [9]. However, a later study has shown that such threat could not be reproduced in more realistic localization experiments [17] where the traffic sign is observed from a variety of viewing directions. This is an important empirical finding since adversarial examples are not endemic to deep learning [22, 19]. In fact, virtually all existing vision systems based on learning (either shallow or deep) are vulnerable to adversarial attacks. Many of these systems will have to be upgraded in order to avoid successful exploits which are likely to arise in near future. Several recent papers offer interesting solutions to this problem [hinton15arxiv, cisse17icml]. Some of them are able to learn on unannotated input images which implies they could be used to support semi-supervised learning [20]. A recent defensive approach achieved almost complete resistance on CIFAR and MNIST datasets [18].

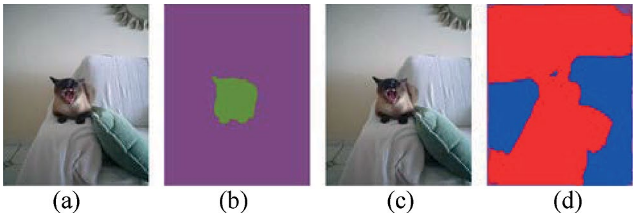

Fig. 5. Original image from Pascal VOC 2007 train (a) and the predictions (b) of our semantic segmentation model (green: cat, purple: background). Adversarial image (c) and the predictions (d) of the same model (red: dog, blue: sofa).

The study of adversarial examples is important even if we disregard the importance of preventing exploits. We know that existing deep models are prone to overfitting due to extremely high capacity [28]. Adversarial examples might lead us towards new regularization techniques that will improve the representation quality and further enhance the accuracy of the predictions.

5. Conclusion

We have reviewed deep convolutional models for semantic segmentation and object localization in natural scenes and presented some of our own contributions in the field. Our experiments [13] were first to confirm the utility of the recently proposed DenseNet architecture [10] for dense prediction in large images. Our model is able to restore the resolution of the dense prediction by blending higher level features at lower spatial resolution with their lower-level higher-resolution counterparts [26]. Such ladder-style blending achieves high spatial accuracy with a very lean upsampling path which significantly relaxes memory requirements and enables real-time processing of large natural images. We are currently able to process 2 Megapixel images (2048×1024) at 13 Hz on a single Titan X GPU with a model that achieves 75% mIoU on the Cityscapes test subset. Our current best result on Cityscapes test is 78.4% mIoU with a multi-resolution forward pass. These figures will improve when we complete our current experiments on the combined training dataset (fine and coarse images).

References

[1] Naveed Akhtar and Ajmal Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. CoRR, abs/1801.00553.

[2] Joan Bruna et al. Intriguing properties of neural networks. ICLR 2014.

[3] Liang-Chieh Chen et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40(4): 834-848 (2018).

[4] Moustapha Cissé et al. Parseval Networks: Improving Robustness to Adversarial Examples. ICML 2017: 854-863.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[6] Piotr Dollár et al. Fast Feature Pyramids for Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 36(8): 1532-1545 (2014).

[7] Dumitru Erhan et al. Visualizing Higher-Layer Features of a Deep Network. ICML Workshop on Learning Feature Hierarchie. 2009.

[8] Mark Everingham et al. The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision 111(1): 98-136 (2015).

[9] Ivan Evtimov et al. Robust Physical-World Attacks on Machine Learning Models. CoRR abs/1707.08945.

[10] Gao Huang et al. Densely Connected Convolutional Networks. CVPR 2017: 2261-2269.

[11] Josip Krapac, Sinisa Segvic, Weakly-Supervised Semantic Segmentation by Redistributing Region Scores Back to the Pixels. GCPR 2016: 377-388.

[12] Ivan Kreso et al. Convolutional Scale Invariance for Semantic Segmentation. GCPR 2016: 64-75.

[13] Ivan Kreso et al. Ladder-Style DenseNets for Semantic Segmentation of Large Natural Images. ICCV Workshops 2017: 238-245.

[14] Alex Krizhevsky et al. ImageNet classification with deep convolutional neural networks. Commun. ACM 60(6): 84-90 (2017)

[15] Laura Leal-Taixé et al. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. CoRR abs/1504.01942.

[16] Wei Liu et al. SSD: Single Shot MultiBox Detector. ECCV 2016.

[17] Jiajun Lu et al. NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles. CVPR Workshop on Negative Results in COmputer Vision 2017.

[18] Aleksander Madry et al. Towards Deep Learning Models Resistant to Adversarial Attacks. CoRR abs/1706.06083.

[19] Michael McCoyd, David A. Wagner: Spoofing 2D Face Detection: Machines See People Who Aren't There. CoRR abs/1608.02128..

[20] Takeru Miyato et al. Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning. CoRR abs/1704.03976.

[21] Chris Olah, Alexander Mordvintsev and Ludwig Schubert. Feature Visualization. Distill, 2017.

[22] A. Ramanathan et al: Adversarial attacks on computer vision algorithms using natural perturbations. ICCC 2017.

[23] Olga Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115(3): 211-252 (2015).

[24] Evan Shelhamer et al. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39(4): 640-651 (2017).

[25] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015.

[26] Harri Valpola. From neural PCA to deep unsupervised learning. CoRR, abs/ 1411.7783, 2014.

[27] Valentina Zadrija et al. Patch-Level Spatial Layout for Classification and Weakly Supervised Localization. GCPR 2015: 492-503.

[28] Chiyuan Zhang et al. Understanding deep learning requires rethinking generalization. ICLR 2017.