COMPARISON OF TRAINING STRATEGIES FOR CONVNETS ON MULTIPLE SIMILAR DATASETS FOR FACADE SEGMENTATION

Matthias Schmitz*, Hai Huang, Helmut Mayer

Institute for Applied Computer Science, Bundeswehr University Munich, Neubiberg, Germany
{matthias.schmitz, hai.huang, helmut.mayer}@unibw.de

Commission II, WG II/6

KEY WORDS: Convolutional Network, Facade Segmentation, Fine-Tuning, Multi-Task Learning

ABSTRACT:

In this paper, we analyze different training strategies and accompanying architectures for Convolutional Networks (ConvNets) when multiple similar datasets are available using the semantic segmentation of rectified facade images as example. Additionally to direct training on the target dataset we analyze multi-task learning and fine-tuning. When using multi-task learning to train a ConvNet, multiple objectives are optimized in parallel. Fine-tuning optimizes these objectives sequentially. For both strategies, the tasks share a common part of the ConvNet for which we vary the depth. We present results for all strategies and compare them regarding the overall pixel-wise accuracy and show that for the special case of facade segmentation there are no significant differences using multiple datasets or not or training a ConvNet with different strategies.

1. INTRODUCTION

Convolutional Networks (ConvNets) have been shown to outperform other techniques in many areas in the last years. Probably one of the most famous applications is image classification. Krizhevsky et al. (2012) started the current Deep Learning era with their impressive results on the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). For semantic segmentation, some authors extended ConvNets, usually designed for image classification (Long et al., 2015) or trained newly developed architectures (Ronneberger et al., 2015). Even if currently the main application area of ConvNets is Computer Vision, they can be applied in many other fields, e.g., speech recognition (Abdel-Hamid et al., 2014) or malware detection (Tobiyama et al., 2016).

One disadvantage of almost all (supervised) Deep Learning methods is that large amounts of training data are needed, because the training of such an architecture entails the optimization of plenty of parameters. While for some tasks big and/or multiple datasets are available, e. g., ImageNet (Russakovsky et al., 2015) for image classification or Cityscapes (Cordts et al., 2016) for semantic segmentation of urban scenes, for many more specific tasks only small or even no datasets exist.

Therefore, many approaches have been developed in recent years to reduce the amount of training data needed for a meaningful optimization. They comprise smarter activation functions (ReLU instead of sigmoid or tanh), (artificial) data augmentation and more sophisticated architectures, such as ResNet (He et al., 2016) or DenseNet (Huang et al., 2017) which introduced skip-connections within sub-blocks of the network.

On the other hand, there are training strategies with accompanying adapted architectures which use additional data in the form of other "similar" datasets with a more or less differing objective. In (Kendall et al., 2018), the same network performs three tasks in parallel: pixel-wise semantic segmentation, object-level instance segmentation and pixel-wise metric depth estimation. All tasks somehow depend on each other, as without knowing what one sees it is hardly possible to determine single objects and knowing objects helps in depth-regression.

In this paper, we analyze different training strategies and accompanying architectures for ConvNets when multiple similar datasets are available using facade segmentation as example.

Facade segmentation has been of increasing interest for different applications, e.g., city planning, augmented and virtual reality or in the movies. While old approaches (Mayer and Reznik, 2006; Reznik and Mayer, 2008) employed multi-view data, most recent approaches use single images for semantic segmentation.

Martinović et al. (2012) proposed a three-layered approach for facade segmentation. A Recursive Neural Network is introduced to obtain probabilistic distributions from an oversegmented facade image in the first layer. In the second layer, a Markov Random Field is used to merge object-level results from specific object-detectors, using the labeling of the first layer. The final layer introduces architectural constraints for more plausible results. Mathias et al. (2016) extended the above approach by optimizing each layer. In (Jampani et al., 2015) an iterative auto-context pipeline is used stacking boosted decision trees for pixel-wise classification.

ConvNets have been introduced for facade segmentation by Schmitz and Mayer (2016). Their work comprises the convolutional part of (Krizhevsky et al., 2012) and a fine-tuned pixel-wise classificator for facades. As the basic architecture includes multiple sub-sampling operations, the resulting resolution was less than the original.

Cohen et al. (2017) employed hard-coded structural constraints on a pixel-wise classification through a dynamic-programming approach. By utilizing symmetries and repetitions their approach is capable to predict occluded facade objects.

Rahmani et al. (2017) used a Structured Random Forest (SRF) for pixel-wise labeling and extended their work in (Rahmani

*Corresponding author

and Mayer, 2018) by adding proposals of a Region Proposal Network to the input of the SRF and by applying deterministic rectangular fitting. A sliding window detector, utilizing a cascade of weak classifiers, is proposed in (Neuhausen et al., 2018).

One particular technique that we analyze in this paper is called fine-tuning or transfer learning: The network is trained on one, usually the larger, less specific dataset (source). Then, the classifier, i.e., the last layer(s) of the network, is replaced by a new one, which is trained on the goal dataset (target).

The other technique which we compare fine-tuning with is multi-task learning: For each task, an independent classifier is defined, but the classifiers share the same basis. This is similar to fine-tuning, but instead of replacing one classifier by another after training the first, both classifiers are trained in parallel.

We show results for both strategies and compare them – regarding the overall pixel-wise accuracy – among each other as well as with direct training, where only a single dataset (target) is used.

In particular, we investigate the problem of semantic segmentation of rectified facade images for two different datasets, namely Graz (Riemenschneider et al., 2012) and ECP (Teboul et al., 2010). For the latter, we use the corrected version (Martinović et al., 2012).

The remainder of this paper is organized as follows: In Section 2, we introduce the basic architecture of our network as well as our modifications, followed by the employed training strategies in Section 3. Details of our experiments are presented in Section 4. We show and discuss results in Section 5 and, finally, conclude the paper with Section 6.

2. NETWORK ARCHITECTURE

The network architecture we have chosen as basis for our experiments is FC-DenseNet56 (Jégou et al., 2017). It is an extension of DenseNet (Huang et al., 2017) for semantic segmentation. DenseNet is defined by a set of dense blocks, in which already extracted features are directly used as input for all subsequent layers within the block. As there is no need for a repeated encoding of information, the training of such a network is more efficient and also possible with smaller amounts of training data.

Figure 1 shows the architecture of the employed network. FC-DenseNet56 consists of an initial convolutional layer, an encoding, transition-down path, a bottleneck dense block, a decoding, transition-up path, and a final convolution layer as pixel-wise classificator. The transition-down path is defined by five consecutive dense blocks. After each dense block, its input and output are concatenated and a transition-down operation is applied. The transition-up part is defined by five consecutive dense blocks as well. A transition-up operation is applied before each dense block and its output is concatenated with the output of the transition-down part at the corresponding depth (skip connection). For details of the implementation of dense blocks, transition-up and transition-down operations we refer to the original paper (Jégou et al., 2017). Our only modification consists of an additional convolutional layer between the last dense block and the final layer, which is included in Figure 1 (red hatched).

For our experiments we defined different versions derived from the basic architecture. The analyzed training strategies, except

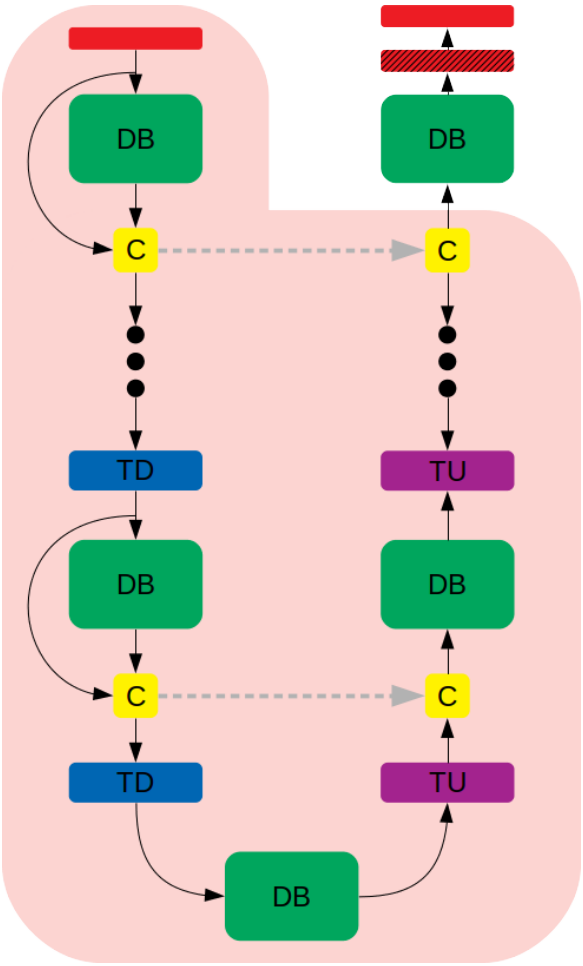
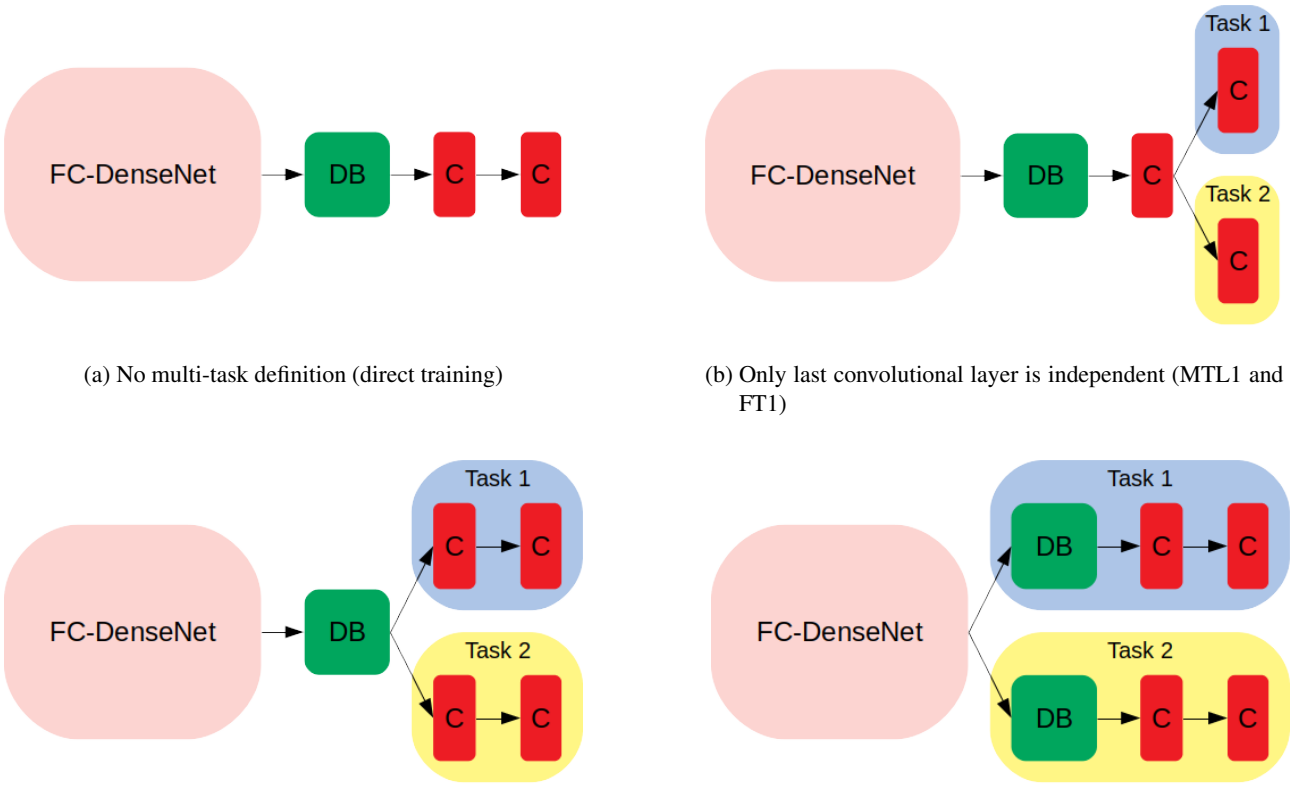


Figure 1. Architecture based on (Jégou et al., 2017). *green:* Dense Block, *red:* Convolution, *blue:* Transition Down, *purple:* Transition Up, and *yellow:* Concatenation. Dashed grey arrows illustrate Skip Connections. The part marked in light red is identical for all our experiments. The hatched convolutional layer is our only architectural modification.

of direct training, require independent classificators (cf. Section 3). To determine which depth of the independent classifiers is optimal, we have trained networks with different split points. Besides direct training (Figure 2a) of the introduced network, we have defined one version with an independent final convolutional layer (MTL1/FT1, Figure 2b), one where both convolutional layers are independent (MTL2/FT2, Figure 2c) and a final one where both convolutional layers as well as the last Dense Block are independent (MTL3/FT3, Figure 2d). The light red parts in Figure 2 are the same as in Figure 1 and the blue and yellow parts correspond to the specific tasks. For both tasks, the number of feature maps is identical, except for the final layer, where it corresponds to the number of classes of the task.

3. TRAINING STRATEGIES

We present our training strategies in the next subsections. During direct training of a network for a single task only the corresponding dataset is used. The other two strategies make use of multiple, different datasets with corresponding independent (but similar) tasks. Both multi-data strategies lead to a common representation of the underlying features by sharing parameters



(a) No multi-task definition (direct training)

(b) Only last convolutional layer is independent (MTL1 and FT1)

(c) Both convolutional layers are independent (MTL2 and FT2)

(d) Last dense block and both convolutional layers are independent (MTL3 and FT3)

Figure 2. The four versions of the proposed network with independent classifiers of different depths. DB: Dense Block, C: Convolutional layer. The light red parts are the same as in Figure 1. The blue and yellow parts correspond to the independent tasks. MTL means Multi-Task Learning and FT Fine-Tuning.

within the basic architecture and have, therefore, the potential to lead to a better generalization.

3.1 Direct Training

The most common way to train a ConvNet is directly on a single task with corresponding data, e.g., pairs of images and labels for image classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). A function has to be defined to compute the loss of the network's output w.r.t. the ground-truth label. Then, the parameters of the network are refined by a gradient-descent algorithm, depending on the computed loss and their gradients, which are computed by back-propagation. Repeated over many iterations and training examples the parameters ideally converge to a (local) optimum.

3.2 Multi-Task Learning

When training a ConvNet with multi-task learning, multiple objectives are optimized in parallel. For instance, Girshick (2015) learned class probabilities (classification) and bounding-box offsets (regression) for an input image and multiple regions of interest. Kendall et al. (2018) trained a network for semantic segmentation, instance segmentation and depth estimation. For each task, an independent part of the network is designed, which follows a common architecture and has its own loss-function.

Contrary to the above approaches, we are not concerned with multiple tasks for the same input image, but take images from two datasets for the training batch in each iteration. We use the sum of the task-specific losses as overall loss, weighted by the relative number of images. When applying back-propagation to compute gradients w.r.t. the parameters of the network, in our case the gradients of the independent parts only depend on the task-specific weights and loss, while the gradients for the shared part are averaged over all examples.

3.3 Fine-Tuning

For fine-tuning a ConvNet to a specific (target) task, it has to be pre-trained on another (source) task. The pre-training corresponds to direct training on the source dataset (cf. Section 3.1). Then, the task-specific part of the network, which usually consists of the last few layers, is replaced and the network is trained on the target dataset. Long et al. (2015) extended image-classification networks (Simonyan and Zisserman, 2014; Krizhevsky et al., 2012; Szegedy et al., 2015) for semantic segmentation and fine-tuned their architecture end-to-end for this task. Fine-tuning can be seen as a special case of multi-task learning, where the training on different tasks is completely asynchronous. To prevent pre-trained layers from overfitting to the new data, the learning rate for these layers is usually reduced. Reducing the learning rate to zero would lead to a feature-extractor followed by a task-specific network.

4. EXPERIMENTS

For both, fine-tuning and multi-task learning, we have trained multiple networks with slightly varied architectures, which differ concerning the position where the independent classifier

starts. By this means, we analyze how deep the shared part of the network should be to achieve the best result. The four versions of the architecture we have used are introduced in Section 2 and shown in Figure 2. Versions (b), (c) and (d) were used for both, multi-task learning and fine-tuning experiments.

4.1 Datasets

Our two datasets consist of rectified and cropped images of facades and corresponding ground-truth annotations.

4.1.1 ECP "Ecole Centrale Paris Facades Database Benchmark 2011" (Teboul et al., 2010) contains 104 rectified and cropped images of single facades in Hausmannian style and corresponding ground-truth annotations with 7 semantic classes (*window, wall, balcony, door, roof, sky, shop*). Because the original annotations were produced by a Hausmannian-style grammar, some labels were imprecise or even wrong. Martinović et al. (2012) provided more precise annotations which we used for our experiments. We split the data into 80 training and 24 test images.

4.1.2 Graz This dataset (Riemenschneider et al., 2012) contains 50 rectified and cropped images of single facades of different styles and corresponding ground-truth labels with 4 semantic classes (*window, wall, door, sky*). The images were generated automatically after extracting a piecewise planar geometry from about 30 different viewpoints. Due to the approximation of the geometry, the projected images contain artifacts. We split the data into 40 images for training and 10 images for testing.

4.2 Training

For all experiments we trained the networks from scratch, using random patches of size 192×192 pixels which are scaled randomly, independent of their dimension, and randomly flipped horizontally. No further data augmentation was applied. As optimization algorithm we used Adam (Kingma and Ba, 2014) with decoupled weight decay regularization as proposed in (Loshchilov and Hutter, 2017). The learning rate was set to 0.001 and the weight decay to 0.0001. We did not change the learning rate for our fine-tuning experiments, as Adam computes individual learning rates for the parameters by estimating first and second moments of the gradients. For the additional convolutional layer we defined 32 kernels of size 1×1. Each network is trained by minimizing the pixel-wise cross-entropy loss for the specific task and by minimizing the weighted sum of the pixel-wise cross-entropy losses for both tasks for multi-task learning, respectively (cf. Section 3.2).

All networks were trained for 150,000 iterations in total on a single GeForce GTX 1080 Ti. The fine-tuning experiments were trained for 100.000 iterations on the source dataset and 50,000 iterations on the target dataset. Training time was around 10 hours per experiment. We set the batch-size to 8 for direct training and the fine-tuning experiments. For the multi-task learning experiments, the batch-size was 12 in total, with 8 examples from the ECP dataset and 4 examples from the Graz dataset, as the former contains twice as much data.

5. RESULTS AND DISCUSSION

In Tables 1 and 2 we show quantitative results for our experiments for the two datasets ECP and Graz, respectively. For comparison we use overall accuracy, i.e., the ratio of correctly classified pixels to all pixels (we ignored pixels that are labeled as void). Surprisingly, all trained networks achieved very similar results for each dataset.

Version	Training accuracy	Test accuracy
DT	.9769	.9323
MTL1	.9687	.9341
MTL2	.964	.9353
MTL3	.6436	.9345
FT1	.9723	.9353
FT2	.9569	.93
FT3	.971	.9319

Table 1. Results after 150,000 iterations for different versions of the architecture (Figure 2) for target dataset ECP. DT: Direct Training, MTLd: Multi-Task Learning, and FTd: Fine-Tuning. d denotes the depth of the independent part.

Version	Training accuracy	Test accuracy
DT	.9812	.9323
MTL1	.9772	.9335
MTL2	.976	.933
MTL3	.9819	.9335
FT1	.9849	.9335
FT2	.9861	.9305
FT3	.9869	.9365

Table 2. Results after 150,000 for different versions of the architecture (cf. Table 1) for target dataset Graz.

The best overall test accuracy on ECP was 0.9353 (MTL2 and FT1) and the worst 0.93 (FT2), with an absolute difference of 0.0053 and a relative difference of 0.57%. For Graz the best overall test accuracy was 0.9365 (FT3) and the worst 0.9305 (FT2), with an absolute difference of 0.006 and a relative difference of 0.64%.

The quantitative overall results show no significant differences, neither between training on a single dataset and training on both datasets, nor between fine-tuning and multi-task learning. Furthermore, the results do not suggest which depth of the independent part is optimal, as there is no clearly observable pattern in the relative ranking.

Thus, we additionally analyze the results of the different versions for the individual classes based on the Intersection over Union (IoU):

$$IoU = \frac{TP}{TP + FP + FN} \tag{1}$$

with TP: True Positive, FP: False Positive, and FN: False Negative.

Table 3 gives the IoU for individual classes for both datasets. The class *door* shows the largest deviation for both datasets, which we assume to be caused by the lowest relative occurrence of *door* in the training data. Especially MTL3 produces clearly better results for this class compared to direct training for both, the ECP and the Graz dataset. Our interpretation is that training a ConvNet only benefits from additional data if there is a gap between the diversity in the original data and the complexity in the represented classes. In our experiments, this only holds for the class *door*, for which the number of training examples is low (only about one door per facade) and the possible appearance varies widely in terms of shape, material or color.

In Figures 3 and 4 we present qualitative results for ECP and Graz, respectively. From left to right we show: image, ground-truth annotation, results from direct training, MTL3 and FT3.

As our approach does not include post-processing and we did not consider symmetries or straight edges in our loss-function, the results for individual objects are "ragged". Most of the windows, for instance, are not perfectly rectangularly shaped and especially for the first two examples in Figure 4, there is no straight boundary between *sky* and *wall*. However, this also holds for the image. Therefore, it is quite interesting that, for the second example of Figure 4, the pointed gable is mostly ignored and classified as sky.

Some of our results show rudimentary objects at the left and right image margins (e.g., a few pixel wide *balcony* on the right side of the lower row of balconies in the first example of Figure 3). While in the result of direct training in the third example of Figure 4 one window in the lowest row is almost totally missing, MTL3 detected a window in the second example, that even is not annotated in the ground-truth data.

Including post-processing, like rectangular fitting, could improve qualitative results, yet, it is not clear if quantitative results would become better, as both datasets are not annotated perfectly correctly.

Although we could not show that any specific training strategy is significantly more preferable than the others, we note that our results are comparable to other state-of-the-art approaches, even without any pre- or post-processing (cf. Table 4). Regarding the overall accuracy, on ECP data, even our worst result (0.93 with FT2) is better than the best result of the referred approaches (Rahmani and Mayer, 2018). All this leads us to the conclusion, that the limit of accuracy is almost reached for these datasets, due to the way the data was annotated. Particularly, there are arbitrary annotation errors by using fixed shapes (rectangles for windows) or rules that separate different classes with a straight horizontal edge, which lead to annotations that are not precise in terms of pixels and also contradicting.

On the other hand, it seems that a sophisticated ConvNet architecture in combination with a good optimization algorithm is able to produce very good results on such data with low complexity (in terms of the number of semantic classes) as well as low diversity (as the images are rectified and cropped) even with small amounts of training data.

Thus, in summary, there is reason to assume that the datasets are not suitable to demonstrate the differences between various training strategies and accompanying architectures of ConvNets for making use of multiple datasets. Even worse, the results for just direct training are so good, that there is no room for improvement by adding information from an additional dataset.

6. CONCLUSION

We have shown that there are no significant differences when training a ConvNet with different strategies on multiple datasets for semantic segmentation of rectified facade images. Neither fine-tuning nor multi-task learning stood out with regard to pixel-wise accuracy or training time. Surprisingly, even direct training on a single dataset led to comparable results.

Nevertheless, we have shown that our results are comparable to those of other state-of-the-art approaches and concluded, that this might be the reason why there was no improvement possible by making use of multiple datasets.

For future work we, thus, want to analyze the training strategies on more complex data, e.g., different view-points and on more diverging data (aerial and terrestrial images for instance). We still consider the question if a specific training strategy can be recommended for a specific class of tasks and if so, what effect the depth of the task-dependent part has, as relevant.

REFERENCES

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., Yu, D., 2014. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22, 1533-1545.

Cohen, A., Oswald, M. R., Liu, Y., Pollefeys, M., 2017. Symmetry-Aware Façade Parsing with Occlusions. *International Conference on 3D Vision (3DV)*, 393–401.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3213–3223.

Girshick, R., 2015. Fast R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q., 2017. Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269.

Jampani, V., Gadde, R., Gehler, P., 2015. Efficient Facade Segmentation using Auto-Context. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1038–1045.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The One Hundred Layers Tiramisu: Fully Convolutional Densenets for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1175–1183.

Kendall, A., Gal, Y., Cipolla, R., 2018. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7482–7491.

Kingma, D. P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NIPS)*, Curran Associates, Inc., 1097–1105.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully Convolutional Networks for Semantic Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440.

Version	ECP								Graz				
	Window	Wall	Balcony	Door	Roof	Sky	Shop	Avg.	Window	Wall	Door	Sky	Avg.
DT	.764	.893	.861	.753	.82	.945	.882	.846	**.8**	.889	.754	.557	.75
MTL1	.779	.898	.849	.749	.818	.946	.888	.847	.784	.889	.716	.621	.753
MTL2	.776	**.9**	.852	.766	.822	.946	**.895**	.851	.798	**.89**	.714	.574	.744
MTL3	**.781**	.896	.856	**.818**	**.828**	.949	.878	**.858**	.794	.885	**.809**	.601	**.772**
FT1	.777	.898	**.863**	.772	.823	**.95**	.886	.853	.792	.886	.777	.605	.765
FT2	.775	.892	.855	.742	.801	.938	.886	.838	.793	.883	.781	.569	.757
FT3	.774	.893	.848	.784	.825	.943	.877	.849	.798	.889	.779	**.622**	**.772**
MEAN	.775	.896	.855	.769	.82	.945	.881	.854	.794	.887	.761	.593	.759
STD	.005	.003	.006	**.026**	.009	.004	.01	.01	.005	.003	**.036**	.026	.011

Table 3. IoU for individual classes for our versions of the ConvNet (cf. Table 1) on both datasets.

Class	[1]	[2]	[3]	[4]	[5]	DT	MTL1	MTL2	MTL3	FT1	FT2	FT3
Window	.823	.78	.87	.804	.786	.853	.866	.876	.869	.84	.846	**.881**
Wall	.929	.89	.91	.915	.935	.946	.952	.941	.947	.952	**.953**	**.953**
Balcony	.893	.87	.92	.864	.892	.921	.907	.905	.926	.916	**.935**	.9
Door	.813	.71	.79	.795	.892	.891	.864	.856	.902	.916	.831	**.929**
Roof	.892	.79	.91	.91	**.93**	.922	.917	.918	.927	.925	.895	.895
Sky	.982	.96	.97	.962	.972	**.987**	.979	.985	.982	.984	.984	.985
Shop	.932	.95	.96	.951	.963	.931	.937	**.973**	.926	.921	.902	.911
Average	.895	.852	.904	.886	.91	.922	.917	.922	**.926**	.922	.907	.922
Overall	.914	.88	.918	.902	.922	.932	.934	**.935**	**.935**	**.935**	.93	.932

Table 4. Results on ECP dataset from: [1] (Jampani et al., 2015), [2] (Mathias et al., 2016), [3] (Cohen et al., 2017), [4] (Rahmani et al., 2017), [5] (Rahmani and Mayer, 2018), and from our versions of the employed ConvNet (cf. Table 1).

Loshchilov, I., Hutter, F., 2017. Decoupled Weight Decay Regularization. *arXiv e-prints*, arXiv:1711.05101.

Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L., 2012. A Three-Layered Approach to Facade Parsing. *European Conference on Computer Vision (ECCV)*, 416–429.

Mathias, M., Martinović, A., Van Gool, L., 2016. ATLAS: A Three-Layered Approach to Facade Parsing. *International Journal of Computer Vision (IJCV)*, 118, 22–48.

Mayer, H., Reznik, S., 2006. MCMC Linked with Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences. *ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI-3, 130–135.

Neuhausen, M., Obel, M., Martin, A., Mark, P., König, M., 2018. Window Detection in Facade Images for Risk Assessment in Tunneling. *Visualization in Engineering*, 6.

Rahmani, K., Huang, H., Mayer, H., 2017. Facade Segmentation With a Structured Random Forest. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, 175–181.

Rahmani, K., Mayer, H., 2018. High Quality Facade Segmentation based on Structured Random Forest, Region Proposal Network and Rectangular Fitting. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2, 223–230.

Reznik, S., Mayer, H., 2008. Implicit Shape Models, Self-Diagnosis, and Model Selection for 3D Facade Interpretation. *Photogrammetrie - Fernerkundung - Geoinformation*, 3, 187-196.

Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H., 2012. Irregular lattices for complex shape grammar facade parsing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1640–1647.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115, 211-252.

Schmitz, M., Mayer, H., 2016. A Convolutional Network for Semantic Facades Segmentation and Interpretation. *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3, 709–715.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints*, arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.

Teboul, O., Simon, L., Koutsourakis, P., Paragios, N., 2010. Segmentation of Building Facades Using Procedural Shape Priors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3105–3112.

Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., Yagi, T., 2016. Malware Detection with Deep Neural Network Using Process Behavior. *IEEE Computer Software and Applications Conference (COMPSAC)*, 2, 577–582.

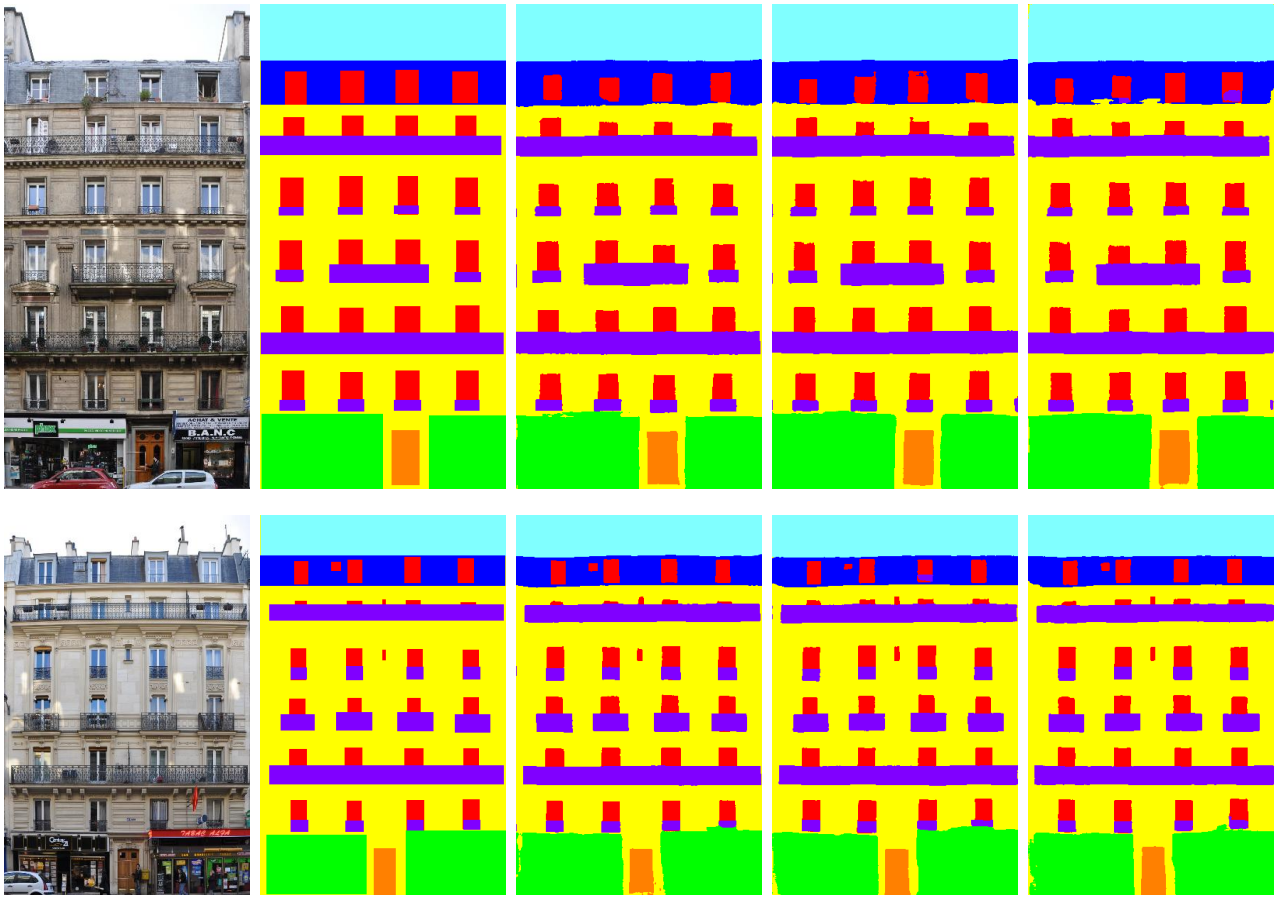


Figure 3. Results for images from ECP dataset. From left to right: image, ground truth, direct training, MTL3, FT3.

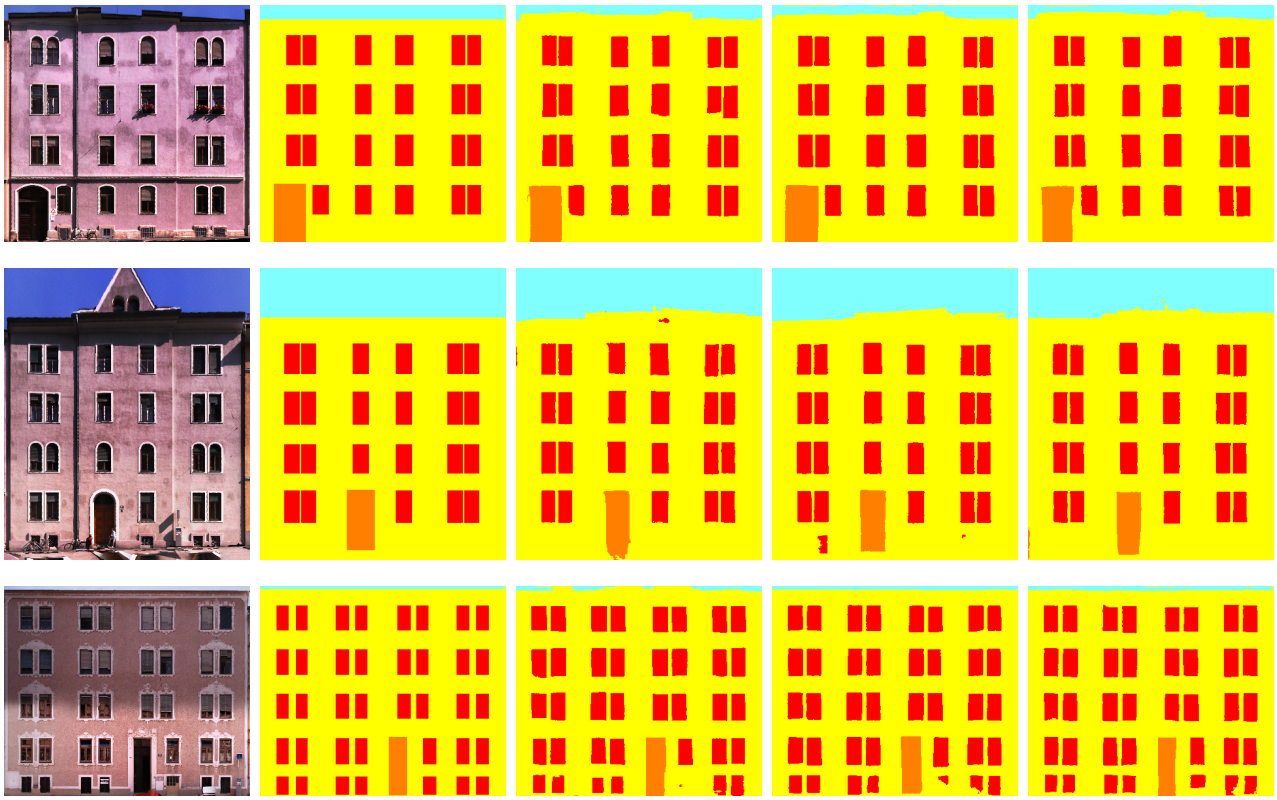


Figure 4. Results for images from Graz dataset. From left to right: image, ground truth, direct training, MTL3, FT3.