Semantic Segmentation for Urban Scene Understanding

Djordje Batic
Faculty of Technical Sciences, University of Novi Sad
Novi Sad, Serbia
djordjebatic@gmail.com

Milica Skipina
Faculty of Technical Sciences, University of Novi Sad
Novi Sad, Serbia
skipinamilica@gmail.com

***Abstract*—In order to achieve a reliable and trustworthy function of autonomous vehicles, understanding the scene in which they operate is a crucial task. Such scene comprehension implies recognizing instances of traffic participants along with general scene semantics. In this paper, we propose an architecture which leverages transfer learning and semi-supervised learning approaches in order to create accurate pixel-level semantic labels.**

We focus on examining the predictive performance of encoder-decoder based architectures trained on ResNet and Inception based encoders, as well as novel encoder pretrained to reconstruct urban scene images. Experimental results show that we can achieve a mean IoU value of 0.626 for the categorical per-pixel semantic labeling task.

***Index Terms*—semantic segmentation, machine learning, neural networks, scene understanding**

I. Introduction

Semantic segmentation is the task of predicting dense per-pixel semantic labels of visual scenes. Pixel-level scene understanding plays an essential role in enabling intelligent behaviour of autonomous vehicles. While humans are able to effortlessly comprehend complex visual scene information, ability to achieve comparable performance in the intelligent system setting remains an open question.

Promising improvements in the field have been made in recent years [1]–[3]. Emergence of convolutional neural networks (CNNs) and availability of large training datasets (e.g. CityScapes [4], KITTI [5] and Mapillary Vista [6]) have made many researches more interested in solving this problem. The object of a scene can be generally categorized into uncountable regions such as sidewalk, sky and road, and countable objects such as cars, bikes and pedestrians. Segmentation of uncountable classes is primarily addressed using the semantic segmentation approach, while segmentation of countable object is approached with instance segmentation task. Proposed solutions used for solving the two separate tasks have diverged, and fundamentally different approaches dominate in each setting [2]. Some of the existing state-of-the-art proposed approaches will be thoroughly examined in the Section II.

In this paper, we propose A novel CNN architecture which provides pixel-level semantic labels for both the countable and uncountable objects in the scene and offers effective urban road scene understanding. The architecture is based on the encoder-decoder network architecture [7]. In order to encode meaningful low level features, we trained the models with pretrained state-of-the-art ResNet [8] and Inception [9] based architectures, while the decoders consisted of convolutional blocks which upsample the encoded representation. As a result, output maps assign one of the eight (8) possible labels to each pixel in the image.

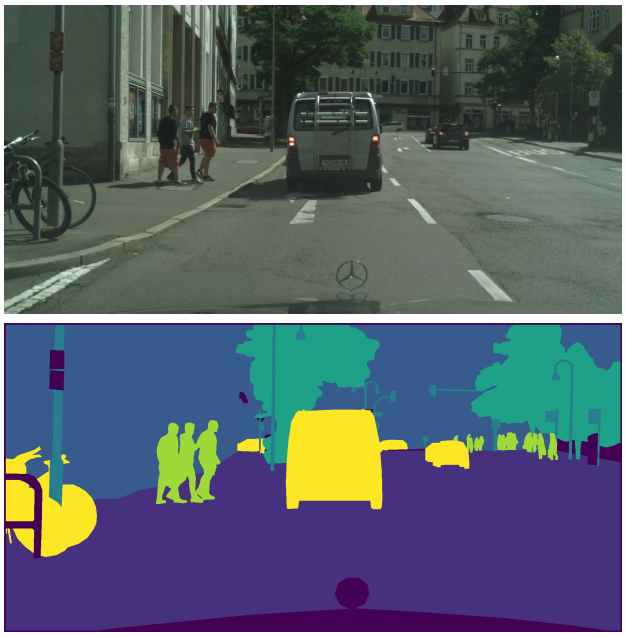

Fig. 1. Top: Input image taken from the training set captured on the streets of Tubingen, Germany. Bottom: False color ground truth label image after transforming the ids to one of eight possible classes.

II. Related Work

In [1] authors presented a novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet. The model is based on encoder-decoder architecture followed by pixel-wise classification layer. The encoder network consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network [10] designed for object classification. SegNet uses all of the pre-trained convolutional layer weights from VGG net as pre-trained weights. Each encoder layer has a corresponding decoder layer. After every max-pooling

layer, max-pooling indices were memorized. These indices later were used for upsampling in the coresponding decoder layers. On CityScapes dataset, SegNet model achived a mean performance IoU value on category-level score of 79.8. Model performed best for the *flat* category (road and sidewalk) with the mIoU value of 97.5, while the worst results with the mIoU value of 43.7 were shown on recognizing *object* category which include poles, traffic signs and traffic ligths.

Another approach for semantic segmentation, based on fully convolutional networks (FCN) is presented in [11]. FCN has been trained end-to-end for pixel-wise prediction from supervised pre-training. Authors fine-tuned existing networks (AlexNet [12], VGG and GoogLeNet [13]) which were adapted into into fully convolutional networks. The resulting networks can predict dense outputs from arbitrary-sized inputs. Fully connected layers from existing nets were transformed into convolution layers. They trained for segmentation by fine-tuning and later added skips between layers to fuse coarse, semantic and local, appearance information. On CityScapes dataset, this model achived a mean performance IoU value on category-level score of 87.5.

Authors in [3] pass multiple scaled images through a network and combine the results using maxpooling or averaging operations. They employ an an attention [14] based approach to combining multi-scale predictions. They propose a hierarchical attention mechanism, which enables roughly 4x more memory efficient training compared to previously proposed approaches. They show that predictions at certain scales are better at resolving particular failures modes, and that the network learns to favor those scales for such cases in order to generate better predictions. They achieve state-of-the-art results in both Mapillary (61.1 mIoU val) and Cityscapes (85.1 mIoU test) datasets while also being memory and computationally efficient, both of which are practical concerns.

Panoptic segmentation approach used in [2] offers joint semantic and instance segmentation. The goal of panoptic segmentation is to assign a unique value, encoding both semantic label and instance id, to every pixel in an image. Authors proposed a two branch approach where semantic segmentation branch is a state-of-the-art segmentation model (e.g., DeepLab [15]), while the instance segmentation branch is class-agnostic, involving a simple instance center regression. Authors report a state-of-art of 84.2% mIoU on the CityScapes test set.

III. CityScapes Dataset

The Cityscapes Dataset focuses on semantic understanding of urban street scenes. A detailed analysis of the dataset, baseline results, and discussions are described in [4]. Dataset contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5,000 frames. We decided to classify each pixel in eight (8) possible classes: **void** ('unlabeled', 'dynamic', 'ground', 'static'), **flat** ('road', 'sidewalk'), **construction** ('bridge', 'building', 'fence', 'garage', 'guard rail', 'tunnel', 'wall'), **object** ('banner', 'billboard', 'lane divider', 'parking sign', 'pole', 'polegroup', 'street light', 'traffic cone', 'traffic device', 'traffic light', 'traffic sign', 'traffic sign frame'), **nature** ('terrain', 'vegetation'), **sky**, **human** ('person', 'rider') and **vehicle** ('bicycle', 'bus', 'car', 'caravan', 'motorcycle', 'trailer', 'train', 'truck'). In order to achieve this, we had to transform the input labels of objects, annotated as "tranId", to their associated category (i.e. cars, bikes, truck etc. to vehicle). Figure 1. shows an example input image and fine segment annotations, where each segment is categorized in one of eight possible classes.

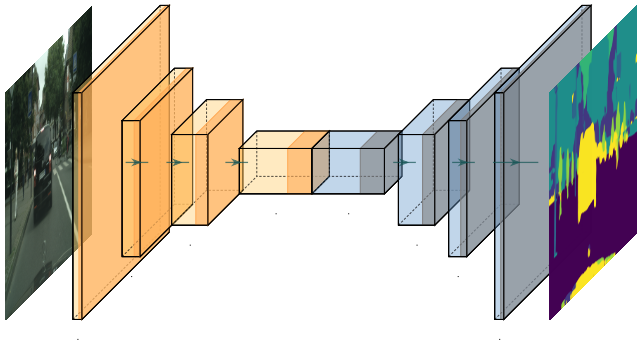

Fig. 2. Overview of our proposed architecture. Given an input image, we use the pretrained ResNet/Inception architecture to yield a down-sampled feature map, then our proposed deep convolutional decoder performs upsampling operation of the received feature map. Finally, semantic pixel-level prediction image is returned as an output of the model.

IV. Methodology

In this section, we describe models we used for building convolutional neural network for semantic pixel-wise segmentation. We used encoder-decoder architectures and built three models with different encoders. For the first two models, the encoders were state-of-the-art models with pretrained weigths: ResNet50 and Inception-V3. For the third model, we used an encoder trained to reconstruct input images from the CityScapes dataset using semi-supervised learning.

All of our approaches follow the general fully convolutional encoder-decoder architecture as show in Figure 2. Motivation behind the proposed solutions will be discussed in further detail below.

A. Encoder

Encoders are one of the core components of convolutional neural network architectures. Extraction of region based visual features, as well as decrease in the computational footprint are one of the key features of encoder-decoder networks. Therefore, we aim to provide a highly accurate representational ability as well as low computational time. In order to achieve this goal, we decided to fine-tune two pretrained state-of-the-art models: ResNet50 and Inception-V3.

ResNet Training a very deep fully connected architecture is a difficult task as the gradients tend to vanish as a function of depth [16], thus degrading the (training) accuracy. The

idea of ResNets is to overcome the degradation problem by helping gradient backpropagation. This is achieved with "shortcut connections" [17] which skip one or more layers in the network. Formally, if a goal of the neural network is to learn the identity mapping:

$$\mathcal{F}(x) = x \tag{1}$$

In the extreme case, we know that as the number of layers increase this mapping approaches zero. To account for this, ResNets aim to learn the mapping:

$$\mathcal{H}(x) = \mathcal{F}(x) + x \tag{2}$$

Intuitively, it would be easier to push the residual to zero, than to fit an identity mapping by a stack of nonlinear layers.

ResNet50 architecture contains four computational blocks with varying number of residual units. We decided to use Bottleneck residuals, as they provide higher computational efficiency compared to Basic blocks. In order to use the architecture as an encoder, we removed the last three pretrained layers (fully connected, average pooling and flatten) and used the rest of the architecture. The output of the model is a 16-times downsampled representation of the input image. We attached a decoder with 4 convolutional blocks, each performing the upsampling operation. As a result, the proposed architecture outputs an original shape (256x256) image containing pixel-level semantic labels. Model was trained in 100 epochs on 10,000 image patches, batch size of 32, learning rate of 0.0003 and learning rate decay rate of 0.5 every 30 epochs. Example predictions of this model can be observed in the Figure 3.

Inception Net Inception v3 is a widely-used image recognition model. Highest quality version of Inception-v3 reaches 21.2%, top-1 and 5.6% top-5 error for single crop evaluation on the ILSVR 2012 classification. Gains in the classification performance tend to transfer to significant quality gains in a wide variety of application domains. This means that architectural improvements in deep convolutional architecture can be utilized for improving performance for most other computer vision tasks that are increasingly reliant on high quality, learned visual features [9]. Authors described a few general principles and optimization ideas that that proved to be useful for scaling up convolution networks in efficient ways.

Authors of the Inception-v3 architecture propose avoiding representational bottlenecks with extreme compression, especially early in the network, and suggest gently decreasing representation size from the inputs to the outputs. They also propose that spatial aggregation can be done over lower dimensional embeddings without much or any loss in representational power and that increasing both, the number of filters per stage and the depth of the network in parallel can contribute to higher quality networks. In comparison with original Inception [13], each 5×5 convolution layer has been replaced by two 3×3 convolution layers which led to reducing the parameter count and cost reduction. Later, they find out that every NxN convolution can be replaced by a 1×N convolution followed by a N×1 convolution which will be the much cheaper solution.

Fig. 3. Left: Input images taken from the validation set. Right: False color prediction images containing pixel-wise labels.

As the pretrained inception-v3 expects tensors with a size of Nx3x299x299, we removed the auxiliary classifiers so that we can use the input images of the same dimensions (256x256) as we used for the other models. Similar to the ResNet encoder described above, we removed the last few layers (fully connected, average pooling, flatten and dropout). Output of this encoder has the dimensions Nx2048x6x6. In order to map this output to the input of the decoder, we added one convolution block which performed upsampling operation with scale factor 3.

B. Decoder

The outputs of our encoders are downsampled feature maps with respect to the input images. Decoders provide a mechanism for upsampling back to the full input resolution. We employed a simple decoder consisting of four convolutional blocks, each performing a 2x upsampling of the image. Block consists of two convolutional layers - 3x3 convolution with stride of 1 and padding of 1, followed by batch normalization, and finally ReLu activation function. Each convolution lowers the channel size by half. Output of the four blocks is passed through a convolutional layer which outputs a *N*x256x256 image, where *N* is equal to number of desired classes. Lastly, softmax activation function is applied, followed by argmax operation in order to create a 1x256x256 image with pixel-level semantic labels. However, model created for image

reconstruction skips this step and outputs a single 3x256x256 reconstructed (RGB) image.

C. Image reconstruction

Besides using pretrained models, another approach we propose is training convolutional autoencoder for reconstructing input images. Encoder and decoder networks are symmetric. Encoder is built by using three convolutional layers followed by batch normalization and ReLu activation function. In order to determine the loss, we calculated the mean squared error of the reconstruction. Results of this model are shown in Figure 4.

After training the image reconstruction model, we detached the pretrained encoder and attached it to the decoder described in the previous subsection in order train a new semantic segmentation model. We were motivated to explore this approach by knowing that the task of encoding the images for reconstruction has a high similarity with the task of encoding the images for segmentation. By doing this, we expect the model weights to settle in the correct distribution useful for encoding representations of the semantic segmentation task. However, upon training the model we observe that it struggles with identifying small, countable objects, particularly street signs. Initially, we suspected an overfitting problem. After applying a 0.5 dropout layer accuracy improved, however the initial poor performance for object class still holds. We suppose that pretraining didn't achieve the desired outcome because it didn't provide robustness to the random initialization of weights thus achieving reduction in model variance. This could perhaps be overcome by leveraging self-supervision and clustering approaches as proposed by Caron et al. [18] Taking this into account, model still performs really well in predicting ground and sky classes.

V. Results

In this section, we present the results of our proposed approach for predicting pixel-level semantic labels. We followed the proposed evaluation technique [4] by removing **void** class predictions from the metric calculation. Our ResNet50 model achieves mIoU of 0.626 while the Inception-v3 achieves mIoU of 0.604. Proposed BatMilNet achieves per-pixel mIoU of 0.386. Results are presented in the Table I.

TABLE I
Results of models for predicting accident severity

class	IoU		
	ResNet50	Inception-v3	BatMilNet
flat	0.845	**0.867**	0.866
construction	0.690	**0.782**	0.555
object	**0.332**	0.178	0.000
nature	0.829	**0.835**	0.676
sky	0.661	0.487	**0.707**
human	**0.503**	0.438	0.191
vehicle	0.518	**0.640**	0.419
mIoU	**0.626**	0.604	0.488

Fig. 4. Top: Input image taken from the validation set. Bottom: Output from convolutional autoencoder.

VI. Conclusion and future work

In this paper, we proposed an architecture for semantic segmentation which incorporates transfer learning approach. We first trained two convolutional encoder-decoder models, where encoders consisted of pretrained ResNet50 and Inception-V3 models, while decoder contained several convolutional blocks which upsample the latent representation to the original (input) image shape and output pixel-wise semantic lebels. Then, a semi-supervised image reconstruction approach was used as a form of network pretraining. We trained a novel encoder network consisting of several convolutional blocks which gradually downsample the image in order to represent it in lower resolution. This encoder network was then used for semantic segmentation by attaching it to the decoder used in previous two architectures.

We show that our approach can achieve satisfactory predictive performance. One of the possible improvements and interesting direction for further research would be in the field of panoptic semantic segmentation. We observe that many of the countable objects (e.g. vehicles and humans) were recognized as a fused object in high density scenes. Overcoming this by incorporating both semantic segmentation for non-countable objects (e.g. ground, terrain, sky) and instance segmentation for countable objects would further improve our solution. Furthermore, with better computational resources we would be able to train on larger sized patches (e.g. 512x1024) which would provide additional boost in accuracy, since our models tend struggle with partially occluded information which appear in higher probability while working with small (256x256) patches. Lastly, data augmentation approaches, such as GANs [19] would be a good step forward in enabling cheaper dataset collection and improvements in the performance.

REFERENCES

[1] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[2] B. Cheng, M. Collins, Y. Zhu, T. Liu, T. Huang, H. Adam, L. Chen, "Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation", 2020

[3] Tao, Andrew & Sapra, Karan & Catanzaro, Bryan. (2020). Hierarchical Multi-Scale Attention for Semantic Segmentation.

[4] Cordts, Marius & Omran, Mohamed & Ramos, Sebastian & Rehfeld, Timo & Enzweiler, Markus & Benenson, Rodrigo & Franke, Uwe & Roth, Stefan & Schiele, Bernt. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. 10.1109/CVPR.2016.350.

[5] A Geiger, P Lenz, C Stiller, and R Urtasun. 2013. Vision meets robotics: The KITTI dataset. Int. J. Rob. Res. 32, 11 (September 2013), 1231–1237. DOI:https://doi.org/10.1177/0278364913491297

[6] G. Neuhold, T. Ollmann, S. R. Bulò and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5000-5009, doi: 10.1109/ICCV.2017.534.

[7] Hinton, G E and Salakhutdinov, R R. "Reducing the dimensionality of data with neural networks." Science 313 , no. 5786 (2006): 504-507

[8] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[10] 1. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv:1409.1556, 2014.

[11] Jian, J., Xiong, F., Xia, W. et al. Fully convolutional networks (FCNs)-based segmentation method for colorectal tumors on T2-weighted magnetic resonance images. Australas Phys Eng Sci Med 41, 393–401 (2018). https://doi.org/10.1007/s13246-018-0636-9

[12] Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. InNIPS, 2012.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842,2014

[14] Vaswani, Ashish, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. "Attention is All you Need." ArXiv abs/1706.03762 (2017): n. pag.

[15] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834-848, 1 April 2018, doi: 10.1109/TPAMI.2017.2699184.

[16] Hanin B, Rolnick D. How to start training: The effect of initialization and architecture. Advances in Neural Information Processing Systems. 2018;2018-December:571-581.

[17] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[18] M. Caron, P. Bojanowski, J. Mairal, A. Joulin. Unsupervised Pre-Training of Image Features on Non-Curated Data

[19] S. Liu, J. Zhang, Y. Chen, Y. Liu, Z. Qin, and T. Wan. Pixel Level Data Augmentation for Semantic Image Segmentation using Generative Adversarial Networks.