Project Report: Reconsidering Decoder Architectures for Road Segmentation

Marco Bagatella, Han Yao Choong, Jonathan Lehner, Matej Sladek (Group: NaN)
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—In this report we present a method for semantic segmentation of high resolution aerial images. We adopt a deep convolutional architecture composed of a state of the art feature extractor and a fully convolutional decoder, on which we study the effects of recently proposed modules such as squeeze and excitation and aggregated residual transformations. Moreover, we ablate the effectiveness of our method and argue its relevance among current approaches.**

I. Introduction

With massive amounts of satellite data being gathered each day, aerial image segmentation has become a widely studied task in machine learning. Automating the extraction of road networks enables important applications such as providing accessibility information for emergency response and accurate maps for GPS navigation [1].

Given a set of RGB aerial images of urban areas, we deal with the task of classifying each 16×16 pixel square as either containing road area or not. A patch is defined to be containing road area if at least 25% of its area is covered by road surface. We achieve this through the intermediate task of pixel-wise image segmentation, for which we assign a label to each individual pixel. We are then able to obtain predictions at a higher resolution, retaining more information and enabling some form of post-processing.

Our model consists of a fully convolutional UNet-like [2] architecture. For our encoder, we rely on a pretrained backbone from the ResNet [3] family with aggregated residual transformations [4], and squeeze and excitation modules [5]. Such architectural modifications have been shown to significantly boost performance in image classification tasks [6] [7], but their application to the decoding path has not been fully explored to the best of our knowledge. For this reason we design a simple and flexible decoder structure and study the effects of such augmentations on it in terms of performance. We find that, while most modifications do not lead to significant improvements, squeeze and excitation modules surprisingly do. We finally boost our model's performance by leveraging ensemble methods and applying morphological post-processing to the predictions.

In part II of this report we mention previous works that our approach builds upon. In part III we explain more in detail the method we propose, while in section IV we present its performance in terms of cross-validated metrics and test scores. Finally, in section V and VI we summarize our findings and present our final considerations on this work.

II. Related Works

The work of Mnih & Hinton (2010) [1] marks one of the first important applications of neural networks to address the task of road detection in aerial images. Prior to the advent of such end-to-end neural network models in visual recognition, much of the previous approaches have been multi-stage, relying on extraction of manually defined features and subsequent classification of regions such as using support vector machines (SVM) [8] [9]. Despite performing well on particular images, such approaches do not generalize well to large real-world datasets [1].

Following Mnih & Hinton (2010) [1], related works [10] and important advances in deep convolutional neural network (DCNN) based visual recognition [11] [12] [13], a class of approaches based on the fully-convolutional network (FCN) architecture proposed by Long et al. (2015) [14] gained prominence due to the consistently high reported performance of its architectural derivatives on a variety of segmentation tasks. One of these derivatives is UNet, proposed by Ronneberger et al. (2015) [2]. UNet aimed to achieve an improved balance between low level details and high level semantic information, i.e. the localization-context trade-off, important for semantic segmentation [2] [14] [15] by the concatenation of feature maps from the encoder to the decoder stage.

In addition to demonstrating the suitability of FCN for segmentation, Long et al. (2015) also demonstrated the effectiveness of using pretrained DCNN classifiers as encoders in FCN models. These classifers serve as pretrained backbones initialized with weights trained on large scale, generalized datasets such as Pascal VOC and ImageNet, and subsequently fine tuned to improve feature extraction for a particular task. This contributed to the consensus that classification performance gains can often transfer to other visual recognition tasks via the high quality features learned by DCNN classifiers [16].

As the originator of the ResNet family from which we obtain our backbone, ResNet was proposed by He et al. (2015) [3] with the aim of easing the training of neural networks with very deep architectures due to vanishing gradients by summing identity mappings from convolutional block inputs with their outputs to preserve activations throughout the course of the network. The success of ResNet, exemplified by its performance at ILSVRC 2015, initiated the development of a series of architectures based on it, culminating in

designs such as ResNeXt, [4] and SE-ResNet, SE-ResNeXt [5]. ResNeXt's central innovation is its use of aggregated residual transformations as a simplification of the split-transform-merge strategy of Inception architectures [16][13] for integration with ResNet. The eponymous squeeze and excitation (SE) blocks in SE-ResNet and SE-ResNeXt on the other hand explicitly model dependencies between feature channels in a convolutional block and adaptively recalibrate feature responses.

III. Models and Methods

A. Datasets and Data Augmentation

Our dataset was partially provided in the context of the course Computational Intelligence Lab at ETH Zurich. It is composed of 100 400×400 RGB training images with binary labels and 94 608×608 RGB test images. Both sets appear to be captured at a resolution of approximately 4 pixels per metre. Because of the small size of the original training set, we incorporate 1800 additional images extracted using Google Maps API. We extract images from both rural and urban locations in the city of Chicago. We assume that the distribution from which these additional images are sampled resembles that of the original dataset.

In order to prepare the images for training we resize them to 384×384, which is the input size of our model. The images are then normalized so that the intensity of each pixel in each channel is in the interval $[0, 1]$. We finally apply augmentations to each image on the fly by randomly flipping it (horizontally and vertically) and rotating it by multiples of a sharp angle.

Test images are treated in a slightly different way: 4 overlapping patches of size 400×400 are cropped from the 4 angles of each image, resized and normalized. From these crops our model outputs 4 predicted masks which are aggregated in a simple way. If a pixel of the original image is only contained in one crop, it receives the label it is given in that crop. In case a pixel is contained into multiple crops (2 or 4), it is assigned a label according to the crop whose center it is closest to. Despite its simplicity, this method does not produce significant artifacts.

B. Model

Our problem is formally a supervised learning task: given a labeled training set $((x_1, y_1), ..., (x_n, y_n))$ drawn from a probability distribution P, the goal is that of learning a parameterized function f_θ that maps any sample x_i *reasonably close* to its label y_i, even for unseen data. Assuming that the training and test data are drawn from the same distribution, this can be formulated as finding

$$\theta^\star = argmin[E_{(x,y)\sim P} L(f_\theta(x), y)]$$

where $L(\cdot)$ is a measure of similarity between ground truth and predictions. In our case, (x_i, y_i) is a pair composed of an RGB aerial image and binary mask labeling the road network, while $L(\cdot)$ is patch-wise accuracy.

In our approach, f_θ is represented by a fully-convolutional deep encoder-decoder architecture. Such a model offers important properties for our task, such as extraction of translation invariant features and integration of information extracted at different resolutions. In keeping with common practice [17] and the approach of Long et al. (2015) [14], our encoder consists of a pretrained residual DCNN. Our choice for feature extraction architecture falls upon SE-ResNeXt101, which incorporates aggregated residual transformations [4], and squeeze and excitation units [5], significantly outperforming ResNet in our experiments.

The top dense layers of the backbone network are removed and the last convolutional block is connected to a decoder composed of 5 blocks. Each block is preceded by an upsampling layer with a 2×2 stride. In order to preserve high resolution information, the decoder is also connected by skip connections to same-resolution layers in the encoder. Encoder features are concatenated to tensors in the decoder after the first 4 upsampling layers. The top of the decoder consists of a single 3x3 convolutional layer with a sigmoid activation function that reduces the number of channels to 1 and produces the predicted mask.

We train our model to minimize binary cross entropy between labels and predictions. Implementation and training details can be found in the following subsection.

The main focus of our work is that of exploring different structures in decoder blocks. In particular we study 5 different designs, labeled as Decoder Block A to E, that are represented in figure 2. Our intention is that of leveraging ideas and techniques used in image classification networks to improve the structure of our decoder.

- Decoder Block A adopts the most common architecture in UNet-like architectures. It is composed of two 3×3 convolutions, each of which is followed by Batch Normalization and a ReLU activation function.
- Decoder Block B introduces a residual connection around the two convolutions in the previously described block.
- Decoder Block C is derived from block A by applying squeeze and excitation [5] to the output of each convolution.
- Decoder Block D substitutes the second convolution in block A by an aggregated residual transformation [4].
- Decoder Block E combines ideas from block C and D, presenting a single aggregated residual transformation as well as squeeze and excitation modules.

The number of filters in convolutional operations starts at 256 in the first decoder block and is halved in each following block.

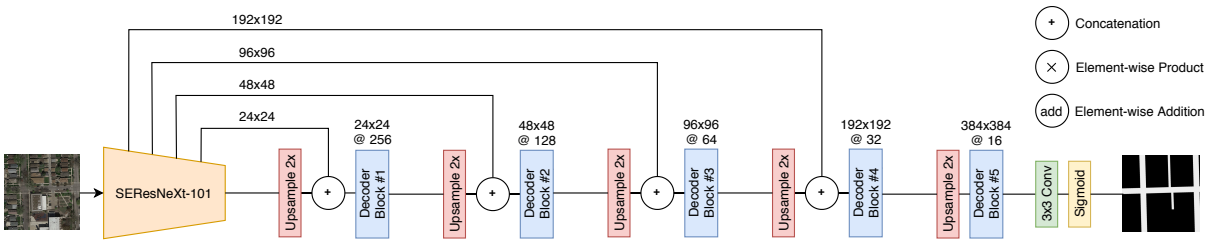

Figure 1. A diagram representing the structure of our network. We also show the symbols we use to indicate the aggregation techniques we deploy.

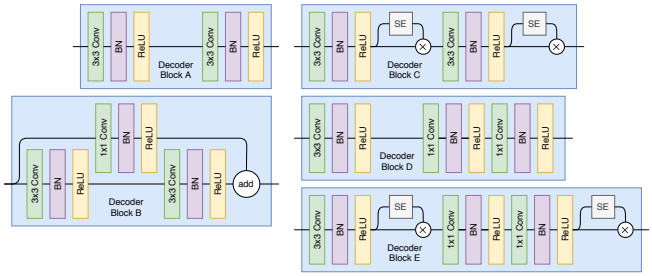

Figure 2. Decoder blocks we experiment on.

C. Implementation and Training

Training data is randomly shuffled and batched with a size of 4, which shows a regularizing effect compared to larger sizes. The backbone is initialized by loading weights pretrained on ImageNet, but they are not frozen to allow for adaptation to a fundamentally different task. We train our network for 25 epochs using Adam as optimizer with a learning rate of 0.0001. Hyperparameters are searched using 5-fold cross-validation.

Our model is implemented in Python relying mainly on the `tensorflow2` library. We trained our models on NVIDIA GeForce GTX 1080Ti GPU's in the Leonhard cluster of ETH Zurich. Training our model takes approximately 110 minutes on such hardware.

D. Post-processing

The output of the network is discretized with a threshold at intensity value 127 for each pixel. We find that occasionally the network misclassifies small paths in parking lots or on sidewalks. We believe this could in part be due to inconsistency in the training data. To remove such false positives as well as noise, we apply a cross-validated number of morphological operations. The mask undergoes 6 iterations of erosion followed by 6 iterations of dilation to obtain the final prediction. In both cases a square kernel of size 3×3 is used.

IV. Results

A. Evaluation and Metrics

The evaluation of models is based on four metrics, namely 16×16 pixel patch-wise accuracy, pixel-wise accuracy, F1 Score and intersection-over-union (IoU). In table III we report 5-fold cross validation estimates for these metrics, as well as test scores reported on Kaggle for models trained on the full dataset. For computational limitations this is the only metric we report for ensemble methods.

B. Experiments

We carried out three groups of experiments to investigate separately the effects of modifying datasets and augmentation procedures (table I), decoder design choices (table II), and finally a comparison of our best approach with two baselines, one of which was introduced in exercise sessions (table III).

V. Discussion

A. Data augmentation

In table I we observe the significant performance gains acquired by training our best model (with Decoder Block C) with additional Google Maps images. Due to inherent class imbalance, this improvement is most manifest in the IoU score, improving by about 0.2 generally. We note that since cross-validation is performed on a different dataset, this represents the model's improvement as a general road segmentation model rather than necessarily an improvement strictly in relation to the original dataset. In fact, we believe that, due to their small sizes, the original training and test dataset are not largely representative of the task.

Data augmentation, on the other hand, has opposite effects when incorporated in training on the original dataset and the combined dataset. The data augmentation applied in this case was solely geometrical, as described in part III. This discrepancy may be attributed to properties of the combined dataset such as the greater prevalence of grid-like road layouts and greater feature variation due to dataset size, causing the trained model to benefit less from generalization afforded by rotation and possibly pick up minor features associated with left-side traffic roads created by reflections, which are in turn missing in validation data.

B. Model Architecture

In our second table we report the effects of adopting different decoder blocks, as described in section III. We find that adding residual connections in our decoder blocks has a slightly detrimental effect on performance. This might be

Additional Data	Random Augmentation	Patch-wise Accuracy	Accuracy	F1 Score	IoU
×	×	0.926 ± 0.012	0.934 ± 0.008	0.852 ± 0.019	0.507 ± 0.040
×	✓	0.935 ± 0.008	0.935 ± 0.008	0.856 ± 0.015	0.506 ± 0.083
✓	×	**0.963 ± 0.001**	**0.957 ± 0.001**	**0.857 ± 0.002**	**0.749 ± 0.003**
✓	✓	0.956 ± 0.001	0.951 ± 0.002	0.835 ± 0.003	0.703 ± 0.006

Table I
EFFECT OF DATA AUGMENTATION ON PERFORMANCE.

Decoder Block	Residual Connection	SE	ART	Patch-wise Accuracy	Accuracy	F1 Score	IoU
A	×	×	×	0.961 ± 0.001	0.956 ± 0.001	0.855 ± 0.004	0.745 ± 0.006
B	✓	×	×	0.961 ± 0.001	0.955 ± 0.002	0.851 ± 0.005	0.739 ± 0.008
C	×	✓	×	**0.963 ± 0.001**	**0.957 ± 0.001**	**0.857 ± 0.002**	**0.749 ± 0.003**
D	×	×	✓	0.962 ± 0.001	0.956 ± 0.002	0.854 ± 0.002	0.744 ± 0.004
E	×	✓	✓	0.962 ± 0.002	0.957 ± 0.002	0.856 ± 0.003	0.746 ± 0.005

Table II
PERFORMANCE OF DIFFERENT DECODER BLOCKS.

Model	Patch-wise Accuracy	Accuracy	F1 Score	IoU	Test score (Kaggle)
Patch-based Logistic Regression	0.654 ± 0.017	0.654 ± 0.017	0.392 ± 0.008	0.244 ± 0.006	0.76954
UNet with ResNet101 backbone	0.947 ± 0.002	0.946 ± 0.002	0.806 ± 0.006	0.666 ± 0.008	0.89696
Our best model	**0.963 ± 0.001**	**0.957 ± 0.001**	0.857 ± 0.002	0.749 ± 0.003	0.92409
Our best model with post-processing	0.961 ± 0.001	0.945 ± 0.002	**0.861 ± 0.002**	**0.756 ± 0.004**	0.92949
Ensemble of our best model	-	-	-	-	0.92701
Ensemble of our best model with post-processing	-	-	-	-	**0.93124**

Table III
COMPARISON WITH DIFFERENT APPROACHES, AS PRESENTED IN SECTION IV

due to the fact that the decoder itself is a shallow architecture in which it is not as hard for gradients to propagate.

We observe that explicitly modeling channel-wise relationships via squeeze and excitation contributes positively to the performance instead. Moreover, SE modules have little to no impact on the training time. To the best of our knowledge this is not a widely used decoder architectural augmentation, but its effectiveness here and in Rundo et al. (2019) [18] may suggest its general utility in DCNN based segmentation models, due to its role in accentuating the relative importance of informative and uninformative features, especially in deeper parts of networks [5]. Among its many properties, one that may be most relevant to our model is the SE mechanism's incorporation of global spatial information to complement spatially constrained convolution operations. The per channel embeddings formed by global average pooling in SE blocks are proportional to relative feature importance in a spatially-agnostic way and this information is then used to recalibrate the magnitude of channel activations, boosting informative features over uninformative ones. The effectiveness of SE may therefore suggest the presence of imbalance in the informative power of features in the decoder, which may point to the presence of some redundant feature channels caused by overparameterization.

C. Comparison with baselines

In table III we present the performance of our best model using Decoder Block C (SE) in the decoder, accompanied by additional data and no random augmentations. Our model vastly outperforms simpler approaches such as logistic regression. This also holds when we compare it to our second baseline, represented by a UNet using ResNet101 as a backbone and Decoder Block A. Post-processing balances out the predictions and has a major impact on the test set, as we believe it mitigates the greater amount of noise caused by the discrepancies between training and testing distributions. We finally present the performance of an ensemble of 10 of our models, that is characterized by lower variance and even better accuracy.

D. Further Work

In the context of this project we also extensively researched multi-task learning as a way to feed more information in the model using contour or distance maps and boost its performance. Despite not being able to produce significant results, we believe this could be an interesting direction to explore. Another interesting approach consists of joint training with an adversarial discriminator.

VI. SUMMARY

We explored the effects of varying decoder block architectures on the performance of a road segmenting UNet-like model. We found that a decoder block structure consisting of two 3×3 convolutional layers, each followed by a SE operation, provided optimal performance for our task. This, coupled with morphological post-processing and ensemble methods, allowed our model to reach the top of the leaderboard in the Kaggle competition for the course Computational Intelligence Lab at ETHZ in 2020.

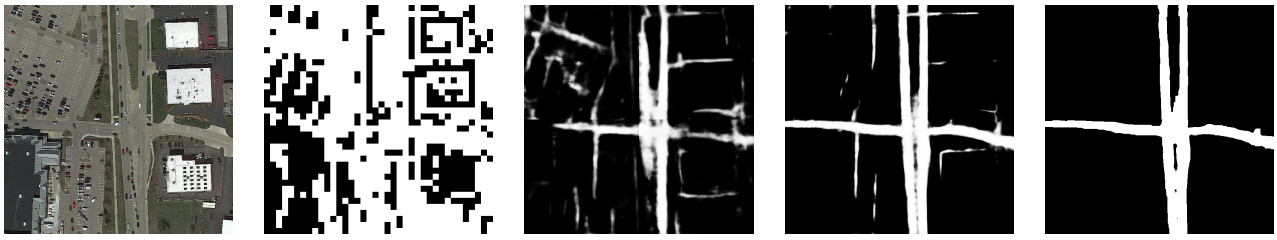

Figure 3. Prediction for a random test image. From left to right: input image, provided baseline (Patch-based Logistic Regression), our internal baseline (UNet with ResNet101 backbone), our best model, our best model with post processing.

REFERENCES

[1] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *European Conference on Computer Vision*. Springer, 2010, pp. 210–223.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[6] W. Han, R. Feng, L. Wang, and L. Gao, "Adaptive spatial-scale-aware deep convolutional neural network for high-resolution remote sensing imagery scene classification," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 4736–4739.

[7] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[8] M. Song and D. Civco, "Road extraction using svm and image segmentation," *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 12, pp. 1365–1371, 2004.

[9] X. Huang and L. Zhang, "Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines," *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 1977–1987, 2009.

[10] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proceedings of the 29th International conference on machine learning (ICML-12)*, 2012, pp. 567–574.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

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

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[15] L.-C. 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," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[17] A. Karpathy, "Stanford university cs231n convolutional neural networks for visual recognition," 2016.

[18] L. Rundo, C. Han, Y. Nagano, J. Zhang, R. Hataya, C. Militello, A. Tangherloni, M. S. Nobile, C. Ferretti, D. Besozzi *et al.*, "Use-net: Incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets," *Neurocomputing*, vol. 365, pp. 31–43, 2019.