Land Cover Classification With Superpixels and Jaccard Index Post-Optimization

Alex Davydow
Neuromation OU
Tallinn, 10111 Estonia
alexey.davydov@neuromation.io

Sergey Nikolenko
Neuromation OU
Tallinn, 10111 Estonia
snikolenko@neuromation.io

Abstract

In this work, we consider the land cover classification task of the DeepGlobe Challenge. This task features the largest available labeled dataset for satellite imagery segmentation. We propose an approach to this problem where standard neural network image classification models are augmented by superpixel extraction and postprocessing that aims to directly optimize the average Jaccard index.

1. Introduction

Satellite imagery is a vast, critically important, and greatly underutilized class of data. Government agencies such as NASA or ESA and companies such as DigitalGlobe [6] have access to terabytes of satellite images that can provide critical data for agriculture, urban planning, sustainable development, early detection and prevention of emergencies and natural disasters, and much more.

Satellite imagery has not yet become the target of much research in computer vision and deep learning. There are few large-scale publicly available datasets, and data labeling is always a bottleneck for segmentation tasks. The DeepGlobe Challenge at CVPR 2018 is designed to bridge this gap, bringing high-quality and at the same time labeled satellite imagery. At the same time, the data in DeepGlobe is still limited, and the labeling is far from perfect—just like it most probably will be in a real life human labeling effort.

In this work, we present a segmentation model for the land cover classification task, one of the competitions in the DeepGlobe Challenge [5]. The main characteristic features of our solution is that it does not rely on latest deep learning architectures but rather introduces several ideas that stem from classical computer vision, including superpixels and postprocessing with direct optimization of the average Jaccard index; we believe that these ideas can significantly improve the results of even state of the art deep learning models.

	Class	RGB	Description
0	Urban land	0,1,1	man-made, built up areas with human artifacts
1	Agriculture land	1,1,0	farms, any planned (i.e. regular) plantation, cropland, orchards, vineyards, nurseries, and ornamental horticultural areas; confined feeding operations
2	Forest land	0,1,0	any land with tree crown density plus clearcuts
3	Water	0,0,1	rivers, oceans, lakes, wetland, ponds
4	Barren land	1,1,1	mountain, land, rock, dessert, beach, no vegetation
5	Rangeland	1,0,1	any non-forest, non-farm, green land, grass
6	Unknown	0,0,0	clouds and others

Table 1. Descriptions of the seven classes in the dataset.

2. Dataset and Evaluation Metric

The input data in the Land Cover Classification task of the DeepGlobe Challenge is satellite imagery collected by a DigitalGlobe satellite with the resolution of 50cm per pixel. The training dataset consists of 803 images, each of size 2448×2448 pixels, in 24-bit JPEG format, and the validation dataset consists of 171 images of the same size and format.

Every image has a ground truth segmentation mask that contains different kinds of land covers labeled with RGB masks. Table 1 summarizes the class labels that are used throughout the dataset. There are $|C| = 7$ classes in total, including one background class "Unknown".

One additional feature of the dataset that makes the competition's problem even harder is that the ground truth labeling quality is not perfect. Some of the segmentation masks are incomplete; many ignore some of the terrain details. This is an intentional feature that makes the problem harder and more realistic (real life handcrafted segmentation is, alas, also often imperfect).

The Land Cover Classification task evaluated the quality of submission with a standard metric for segmentation quality, the pixel-wise mean Intersection over Union (mIoU), or

Jaccard index:

$$\mathrm{mIoU} = \frac{1}{6}\sum_{c=1}^{6} \mathrm{IoU}_c,$$

where IoU_c for a specific class c is defined as $\mathrm{IoU}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c}$, i.e., the ratio of the number of true positive pixels in class $c \in C$ summed over the entire dataset and the total number of pixels in the union of the ground truth mask and the predicted mask. In the competition, the mIoU metric was computed by averaging over the six "meaningful" classes, all except the "Unknown" class.

3. Methods

3.1. Problems and the basic idea

The classification problem in the land cover classification task has several features that make it especially difficult. First, the labeling is (intentionally) bad, with many errors that reflect real life problems that occur with labeling. Second, not only are the classes imbalanced, but the balance is completely different in the training and validation set; e.g., one of the classes has share 0.6 in the validation set, and another only 0.006. Moreover, one of the classes ("rangeland") has especially incomplete labeling, so the mean accuracy one can expect from the classes varies greatly.

As is common in segmentation problems, the objective function (Jaccard index) does not match the maximal likelihood objective function (cross-entropy). There are two possible approaches to this mismatch. First, one could devise new objective functions that more directly optimize the Jaccard index; previous work in this direction included the Jaccard hinge loss [3] and the recently developed Lovász-Softmax loss [4].

Another solution is to perform postprocessing, as done in, e.g., [9]. For many segmentation problems the first approach is preferable, but in this case, since the balance between classes is skewed and very differently skewed in the training and validation sets, it is unlikely to work well, so in this work we tried the second approach, first applying a standard classification network and then using different postprocessing approaches.

3.2. Classification model: superpixels and Haralick features

We trained the basic classifier on the cross-entropy loss function. One important problem with it was that cross-entropy converged very badly due to imperfect labeling in the training set. To alleviate this problem, we propose to unite pixels into superpixels on a larger grid; this approach improves convergence significantly, but obviously introduces a tradeoff: large superpixels mean that fine details get lost in the process.

In our experiments, the best superpixel size proved to be 115×115 pixels. The second parameter of superpixels is their *compactness*, with a tradeoff between compactness and boundary recall studied in [10]. We have tried several algorithms for superpixels and found that the best one in our experiments was simple linear iterative clustering (SLIC) with a ruler of 15 [1]. The classification of superpixels was done by classifying 50×50 squares sampled from the superpixel; a similar approach has been used, e.g., in [2].

We have compared VGG and Inception v3 classifiers trained on the cross-entropy loss. Moreover, to help the classifiers, we added a separate input consisting of Haralick features [8] and the average CIELAB color space features that are known to work well for water segmentation that are input before the first fully connected layer. Figure 1 shows the network architectures we have used and compared. Since we have also added an extra dense layer on top of the Haralick features, we have also compared the Inception v3 architecture with an extra dense layer to make sure that the performance gain indeed can be attributed to Haralick features.

Numerical results of our experiments are shown on Fig. 2. We have found that networks with additional Haralick features converge noticeably better albeit somewhat slower. Note also that the version with Haralick features are less prone to overfitting, while, as expected, simply adding an extra dense layer does not do much at all.

Overall, our results in these experiments show that techniques from classical computer vision can still help even modern deep learning models.

3.3. Postprocessing

After the classifier has been trained, we apply postprocessing to optimize the results for Jaccard index loss rather than cross-entropy (or accuracy). Suppose that we have calibrated the network well and have obtained estimated of class probabilities $p_x(c)$ for every superpixel x in an image. This means that we can estimate the expected number of true positives $\mathrm{TP}(c)$, false positives $\mathrm{FP}(c)$, and false negatives $\mathrm{FN}(c)$ for every class by adding the corresponding probabilities, and this yields the expected Jaccard index J for the current result.

Then we perform the following procedure: for every superpixel, check whether we can increase the expected Jaccard index by flipping the value of this superpixel. This greedy optimization procedure significantly improved the Jaccard index on local validation and on the validation set. Note that another important feature of our approach is that since we are computing probabilities over superpixels rather than individual pixels the postprocessing becomes much faster: there are only ≈ 500 superpixels for an image of size 2048×2048 pixels; doing this greedy postprocessing pixel-wise would be infeasible.

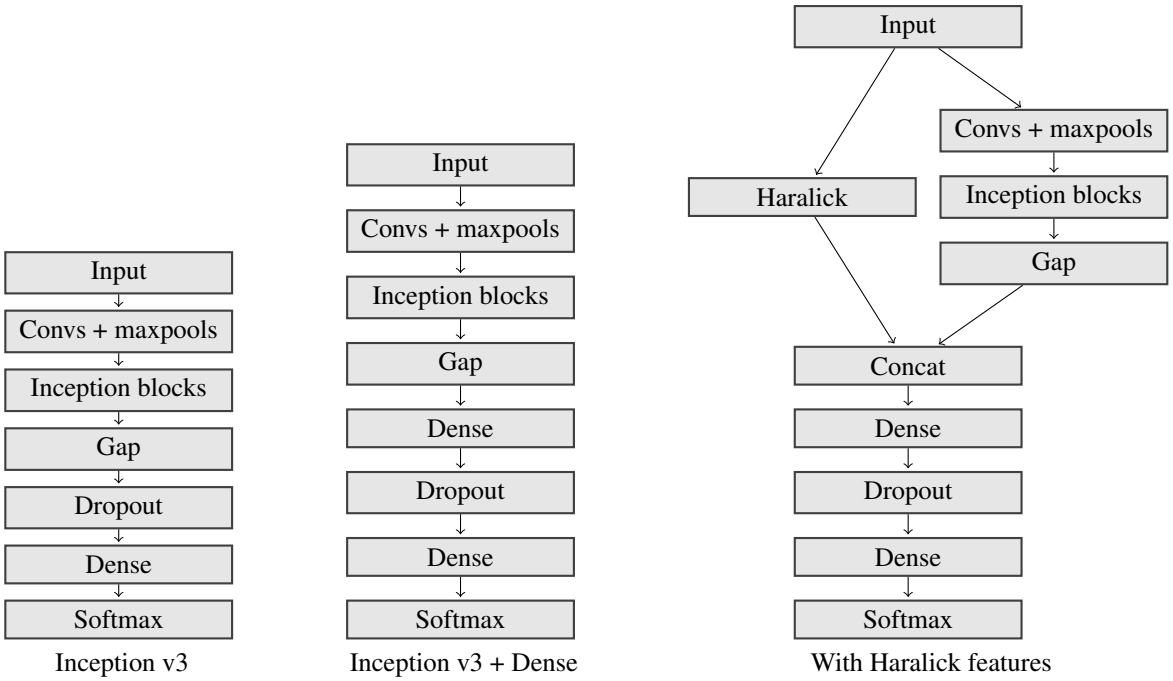


Figure 1. Network architectures used in our solution.

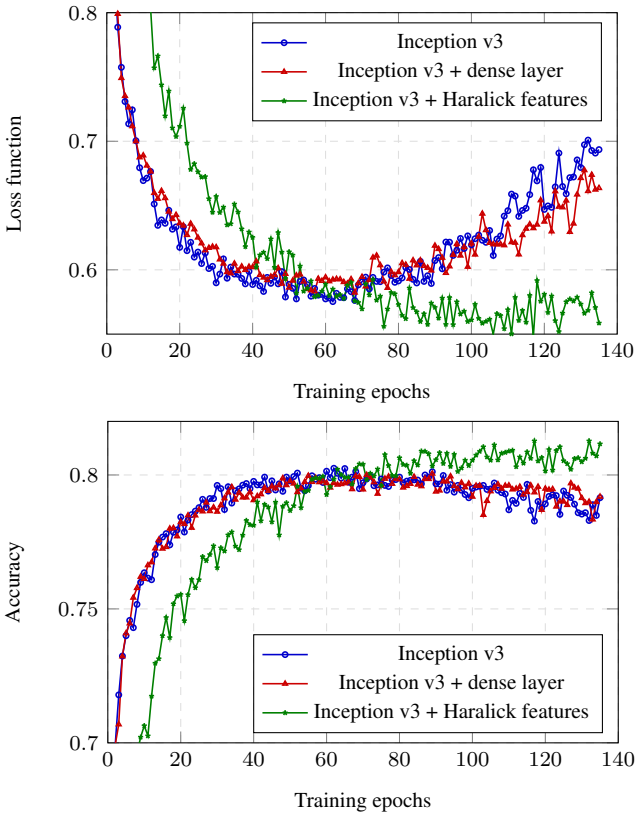


Figure 2. Performance results for the three networks

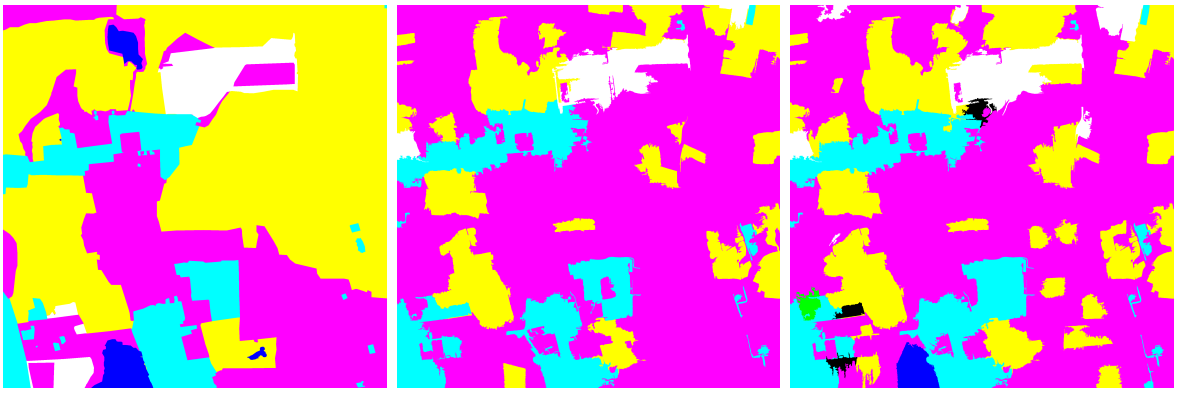
Our postprocessing procedure requires a well-calibrated neural network. We have compared calibration with isotonic regression and temperature-based calibration [7]; in our experiments, isotonic regression significantly deteriorated discrimination and led to worse overall results, while temperature-based calibration did help, so we went with this approach.

3.4. Results

The results of our experiments are summarized in Table 2. The columns show precision, recall, and F1 score of the classifiers (for superpixels) separated into six classes (except "Unknown"), and then show the average Jaccard index for the same classifier (for individual pixels, not superpixels) without postprocessing and with postprocessing. We show results on the local validation set, so effects of postprocessing are relatively small; they are much more pronounced for highly imbalanced classes such as the ones in the leaderboard validation set.

To compute evaluation metrics that would be more indicative of the results on a validation set, we first estimated the relative support of classes on the validation set (with a separate network trained on the cross-entropy loss) and then reweighted the local validation set so that the support of each class is proportional to the leaderboard validation set; these are the results reported in Table 2.

This modification also helped us perform one additional fine-tuning: we tuned weights of the samples in each class



(a) (b) (c)

Figure 3. Sample segmentation results. Left to right: (a) ground truth mask; (b) model result before postprocessing, average Jaccard index 0.317, accuracy 0.572; (c) model result after postprocessing, average Jaccard index 0.393, accuracy 0.584.

Table 2. A summary of experimental results.

Classifier	FT	Precision							Recall							F1 score							Avg Jaccard index	
		0	1	2	3	4	5	avg	0	1	2	3	4	5	avg	0	1	2	3	4	5	avg	w/o post	with post
Pure Haralick		0.53	0.52	0.72	0.93	0.51	0.27	0.54	0.81	0.67	0.86	0.64	0.37	0.06	0.64	0.64	0.59	0.79	0.76	0.43	0.09	0.58		
VGG	No	0.59	0.95	0.86	0.90	0.42	0.40	0.83	0.87	0.78	0.91	0.85	0.63	0.57	0.79	0.70	0.86	0.88	0.87	0.51	0.47	0.80	0.643	0.646
Incept. v3	No	0.65	0.96	0.87	0.78	0.32	0.46	0.84	0.86	0.77	0.90	0.87	0.76	0.64	0.79	0.74	0.85	0.88	0.82	0.45	0.54	0.80	0.646	0.659
Incept. v3	Yes	0.78	0.90	0.89	0.86	0.54	0.54	0.84	0.82	0.91	0.87	0.85	0.52	0.55	0.84	0.80	0.91	0.88	0.86	0.53	0.54	0.84	0.661	0.604

during training so that the F1 score computed above improves as much as possible. In Table 2 we see that this modification (denoted "Fine-tuning") indeed improves local validation results significantly. Note, however, that even after tuning results on the validation leaderboard were much lower than on local validation, 0.4764 vs. 0.66, despite the fact that for tuning we used relative support of the classes estimated on the validation set. We suspect that this effect is due to the properties of the dataset: it appears that the training and validation sets were sampled from different regions with different appearances. In particular, we have noticed that forests inside each dataset are similar to each other but the training set has almost exclusively coniferous forests while the validation set has almost exclusively greenwood. Interestingly, pure Haralick features augmented with the mean color of a superpixel, with no deep neural networks at all, perform not all that much worse than more complicated model (first row of Table 2). As for the results with deep neural networks as classifiers, we see that Inception v3 significantly outperforms VGG, and fine-tuning with weighted input samples also helps improve the results. Note, however, that for the Inception v3 classifier with fine-tuning postprocessing actually makes the result worse; this is due to the fact that fine-tuning imposes completely different calibration to different classes, and temperature-based calibration does not help anymore.

3.5. Sample results

Figure 3 shows sample segmentation results of our model. It shows the image itself, the ground truth segmentation mask, results of the classification model, and results after postprocessing. We see how postprocessing significantly improves the quality in this example.

This example also shows an important feature of our solution: it is able to make use of the "Unknown" class (shown in black); black pixels contribute only to the false negatives and do not yield false positives as an incorrectly predicted class would, so they also contribute to the increase in the average Jaccard index.

4. Conclusion

In this work, we have proposed an approach to land cover classification based on relatively simple standard deep neural networks used as classifiers for superpixels. We have introduced several ideas, both from classical computer vision and recent ideas regarding Jaccard index optimization, that have let us obtain very good results in this challenge.

As for further work, in this solution we have concentrated on extra features and postprocessing, and the deep learning part is standard and even a bit outdated. It is very promising that we were still able to obtain good practical results, and it would be very interesting to think of a way to combine the ideas of this approach with "standard" modern segmentation techniques such as U-Net and similar architectures. We believe that a combination of our ideas shown in this work with better basic deep learning models for image segmentation and classification can further improve state of the art in segmentation for satellite imagery.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, Nov. 2012.

[2] N. Audebert, A. Boulch, H. Randrianarivo, B. L. Saux, M. Ferecatu, S. Lefvre, and R. Marlet. Deep learning for urban remote sensing. *2017 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4, 2017.

[3] M. Berman and M. B. Blaschko. Optimization of the jaccard index for image segmentation with the lovász hinge. *CoRR*, abs/1705.08790, 2017.

[4] M. Berman, A. Rannen Ep Triki, and M. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. 2018.

[5] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. *ArXiv e-prints*, May 2018.

[6] S. L. P. Fred A. Kruse, William M. Baugh. Validation of digitalglobe worldview-3 earth imaging satellite shortwave infrared bands for mineral mapping. *Journal of Applied Remote Sensing*, 9:9–17, 2015.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017.

[8] R. M. Haralick, I. Dinstein, and K. Shanmugam. Textural features for image classification. *Ieee Transactions On Systems Man And Cybernetics*, 3(6):610–621, 1973.

[9] S. Nowozin. Optimal decisions from probabilistic models: The intersection-over-union case. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 548–555, June 2014.

[10] A. Schick, M. Fischer, and R. Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 930–934, Nov 2012.