CS-433 project 2. Challenge: road segmentation

Takuya Ishii Daniel Selmin Matthew Morvan

December 21, 2023

Abstract

Our second project for CS-433 tackles the road segmentation task on satellite imagery. We take the deep learning approach to automatically extract features from the provided images and choose a convolutional neural network based UNet architecture. Our main consideration was the comparison of region-based Generalized Dice Loss (GDE), distance-based Boundary Loss (BL) and the combination of these two. Various data augmentations and post-processing techniques were also considered, even though their contribution to the test F1 score was negligible or detrimental within the scope of our analysis. Ultimately, we report a test F1 score of 0.880 using using GDE, and discuss the generalizability of BL beyond the scope of this specific road segmentation task.

1 Introduction

Achieving an optimal balance between precision and recall is essential to road segmentation tasks, as falsely labeling roads where there aren't, or omitting them where they exist can lead to catastrophic incidents, ranging from navigation errors in autonomous driving, to neglecting vital zones in disaster response planning. Custom-tailored loss functions can be implemented in segmentation tasks in conjunction with convolutional neural network (CNN) models in order to optimize with regards to false positive (FP) and false negative (FN) tradeoffs. Regional loss functions like Generalized Dice loss (GDL), which prioritize pixel-wise accuracy by overlapping segmented areas with the ground truth of training samples, already strive for a harmonious balance between precision and recall by computing the harmonic mean of the two [1]. However, relying solely on regional loss may fall short when the model is trained on a highly imbalanced dataset, where the prevalence of non-road regions can overshadow smaller, yet crucial, road segments.

Methods like Boundary loss (BL) provide a complementary approach, emphasizing the accurate delineation of road contours rather than pixel classification alone [2]. As such, the problem becomes one of optimizing distances as opposed to one of classifying sets as true or false, a particularly beneficial improvement for applications where the precise layout of roads is paramount.

Here, using a small dataset of satellite images and classic data augmentation techniques, we explore a combined loss approach which successively prioritises GDL and BL during training. While we initially observe better results with GDL (F1 score = 0.880), we discuss the robustness of regional loss methods compared to the proposed combined loss, and suggest that despite combined loss' slightly worst performance on our test-set, it might be more generalizable than classic GDL.

2 Methods

2.1 Dataset

Our dataset consists of 100 satellite images (400x400, RGB) with ground truth road segment labels provided as masks, and 50 images (608x608, RGB) with no labels. We make a 4:1 split to the dataset and create a training set with 80 images and validation set with 20 images, while treating the unlabelled images as a test set.

2.2 Loss Functions

Segmentation approaches based on convolutional neural networks (CNN) are typically trained by minimizing cross-entropy (CE), a measure of the affinity between regions defined by probability softmax outputs of the network and the corresponding ground-truth regions [3]. However, in the case of unbalanced data, CE requires large amounts of data to train, and must be improved with class weighting techniques, the most popular of which is Generalized Dice Loss (GDL), which assigns weights to classes according to the inverse of the class label's frequency [1]. More precisely, the Dice coefficient measures the similarity or overlap between two sets, in this case, the predicted mask for our road, and the ground truth mask defined such that:

$$\text{Dice} = \frac{2 \cdot \text{Intersection}}{\text{Union} + \text{Intersection}}$$

In the context of segmentation, "intersection" represents the number of overlapping pixels between the predicted and ground truth masks, and "union" represents the total number of pixels in both masks.

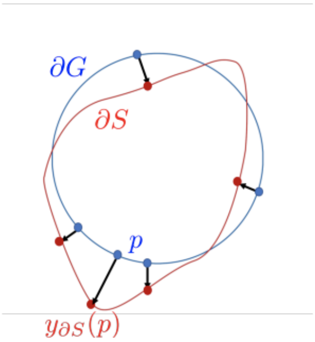

Figure 1: Determining Boundary Change

Boundary loss offers a complementary solution to GDL, which the authors claim can mitigate issues related to regional losses in highly unbalanced segmentation problems [2]. As opposed to GDL which uses unbalanced integrals over regions, BL uses integrals over the interface between regions. More precisely, boundary loss (or surface loss) takes a non-symmetric L2 distance between two object boundaries, and subsequently sums these as a regional integral.

$$\mathrm{Dist}(\partial G, \partial S) = \int_{\partial G} \|\mathbf{y}_{\partial S}(p) - p\|^2 dp$$

The BL is then obtained by summing the linear functions of the regional softmax probabilities output by the network. In other words, once our network has computed our regional integrals we determine the likelihood of each region belonging to a certain shape or contour. Because the information provided by the surface loss is complimentary to that of regional losses like dice loss, we combine dice loss and surface loss as such:

$$\text{Loss} = \alpha \mathcal{L}_{GD}(\theta) + (1 - \alpha)\mathcal{L}_B(\theta)$$

Where α is a value between 0 and 1 that determines the significance of each loss function at different epochs, initially prioritizing GDL and later BL [2].

2.3 Model description

We chose UNet [4] as the backbone architecture of our model while selecting ResNet18 [5] pre-trained on ImageNet [6] as our encoder. By taking advantage of transfer learning [7] with Imagenet, we aimed to accelerate the convergence of our loss functions and therefore reduce our training time. We chose Adam as our optimizer, setting its initial learning rate at 0.001 [8]. With regards to the model's implementation, the computation of our loss functions and the model's training, we relied on `Segmentation Models` [9], `PyTorch Lightning` [10] and `SciPy` [11].

Upon computing the distance map, it should be noted that for slices containing only the background region, we used a zero-distance map, with the assumption that GDL would be sufficient in those cases. Moreover, during training, the value of α was set to 1 at the start of each training, and decreased after each epoch, following a simple scheduling strategy, until it reached the value of 0.01. As such, we weigh the regional loss (i.e., Dice loss) more heavily at the beginning and progressively increase the impact of the boundary loss term at each epoch. This scheduling strategy, proposed by [2], is suggested in order to make the results of the model less contingent on the choice of α while giving consistently similar or better results than a constant value. We also benchmark the performance of this combined loss with:

1. a combined loss where α is reduced by 0.005 at each epoch (a strategy we choose after observing...),
2. the boundary loss, or
3. the dice loss run alone.

Our experiments were hosted on a Google Colab machine equipped with an NVIDIA V100 Tensor Core GPU with 12.7GBs of memory. Note that for questions of computational power, we did no implement cross-validation methods to optimize the hyperparameters of our model. For evaluation purposes, we employ Intersection of Union [12] and F1 score metrics over the dataset.

2.4 Image Augmentation

In order to adjust the image size to the CNN model of our choice and add variability to the images in each epoch, we apply random cropping and use parts of the original image for model training [13]. The new cropping is applied every time a new batch is generated by PyTorch `Dataset` class. On top of random cropping, we consider rotation in [0, 180] degrees, horizontal flipping and colour jittering all randomly to further increase the diversity of our dataset (see Figure 2) [14]. As a control to these transformation experiments, we also create a training/validation set with resizing to 416x416, keeping the original size of the images (400x400) as much as possible while maintaining the compatibility with our CNN model.

2.5 Post-processing

Using `OpenCV` library, we apply a simple morphological transformation called opening [15]. This is a two-fold process applying erosion and dilation consecutively to the input image. Erosion is a sliding kernel operation which replaces the pixel value of

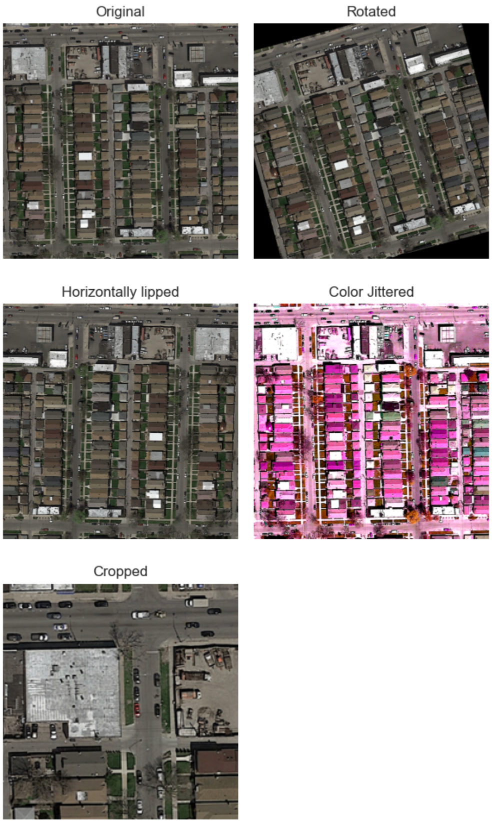

Figure 2: Different transformations applied to our images. During the training, all the transformations can be applied simultaneously.

the anchor point (often at the centre of the kernel) with the minimal value inside the kernel. Dilation is a similar process but the anchor value is replaced by the maximal value inside the kernel. We used 3x3 kernel for both erosion and dilation operations, and each operation was repeated 3 times for erosion and 4 times for dilation.

3 Results

First, we compare the effects of transfer learning on our GDL model (Figure 3) and observe a much faster decrease in our Dice loss when the training is conducted after loading pretrained model weights, demonstrating the effectiveness of ImageNet-based model weights on our road segmentation task.

Second, we compare the effects of different augmentations on our GDL model using F1 scores as our benchmark metric. In testing combinations of cropping, horizontal flipping, resizing, rotations, and photometric distortions, we notice that resizing alone gives us the highest F1 score at 0.9126. However, for the sake of evaluating the effectiveness of our combined loss, we also run our model with these transforms to test its consistency across a more diverse training set compared to GDL.

Third, to evaluate our model, we compare the F1 scores of the models trained with:

1. our combined loss ($\alpha = 0.01$)
2. our combined loss ($\alpha = 0.005$)
3. GDL on its own
4. BL on its own

To achieve the highest F1 score on our test set, we benchmark these loss functions by applying the resizing transform, running the training for 100 epochs and measuring the F1 score on our validation set. The highest F1 scores obtained is with GDL (3.), and we get 0.9073, 0.898, 0.9126 for the first 3 models respectively. Note that we do not evaluate BL here as it is too unstable when implemented on its own (see Figure 4). To evaluate our test set we measure our F1 score according to the EPFL 2023 Road Segmentation competition's metrics, where a value of 0 or 1 is successively assigned to 16x16 patches of our 608x608 test set prediction masks and subsequently compared to the test set's corresponding ground truth masks. With this F1 score we get 0.880 for GDL and 0.877 when adding our post-processing, a slight decrease in the test F1 score.

However, in an attempt to create a more robust model and increase the test F1 score, we also benchmark the different loss functions by training them with all 4 transformations shown in Figure 2 so as to expose its training to samples that are substantially different than the ones it will be tested on. Figure 4 shows the evolution of the F1 scores across this virtually augmented training set for 200 epochs for which we observe an F1 score of 0.7467, 0.7793, 0.7257 and 0.2482 respectively. The EPFL 2023 Road Segmentation F1 scores again render slightly higher scores of 0.836 for our GDL implementation, 0.877 for our combined loss and a final 0.879 score when adding post-processing to our combined loss ($\alpha = 0.005$); in other words, when the training data contains more diverse set of pictures, combined loss ($\alpha = 0.005$) performs better the GDL.

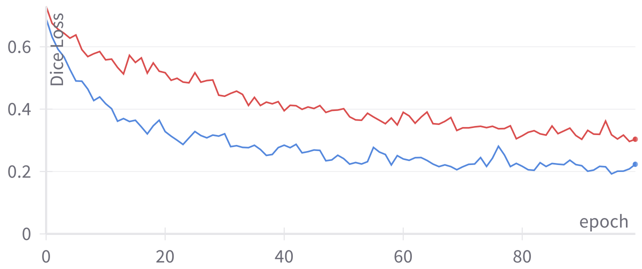

Figure 3: Change in Dice Loss over 100 epochs. The blue line (with Imagenet) shows much faster convergence than the red line (without Imagenet). All image transformations shown in Figure 2 were employed during training.

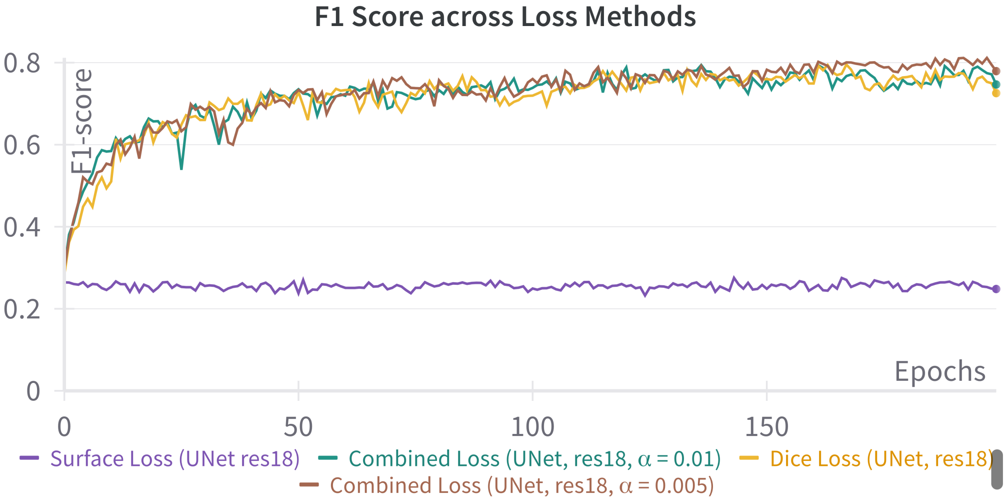

Figure 4: F1 score across loss methods for training set

4 Discussion

Our results show that our model performs best with a GDL function using the ImageNet pre-trained weights. The model scores highest when applying a simple resizing to its train data, rendering an F1 score of 0.9126 on the validation set and 0.880 on the test set. The performance of our combined loss (α = 0.005) and combined loss (α = 0.01) at 0.9073 and 0.898 respectively, portray this method as practically on par with GDL for our task. However, the fact that our F1 score is higher when our model is implemented with only resizing is symptomatic either of ineffective data augmentations (i.e., adding too much variation can make the dataset too different from the original data distribution) or of overfitting on limited data augmentations.

As seen previously, augmentations like cropping [13], flipping and color jittering are common in segmentation tasks [14], and should reflect real-world scenarios where satellite images fed to a road segmentation model can come with different sizes, angles and resolutions; given that our train set is rather small, these initial results could thus indicate overfitting. Moreover, given that combined loss first performs with an F1 score of 0.7793 on its train/validation set and a subsequent F1 score of 0.877 on its test set implies that the variation added during our data augmentation still renders good results. Furthermore, the fact that it performs better than GDL (despite being trained on more differentiated data) by a margin of 0.053 on its training set and 0.041 on its test set indicates that combined loss can potentially capture more variance and thus be more robust and generalizable than GDL. While this hypothesis would have to be tested on larger test sets by using methods including K-fold cross-validation to validate its tolerance to varied inputs, this finding can be explained by the nature of the distance map provided as a target for BL. Because GDL does not use any spatial information, it treats errors equally whether FPs or FNs. However, because BL is based on a distance map of every pixel to the target object, misclassified road areas both small and/or far will be more heavily penalized. In other words, because FPs will be far away from the closest foreground, they will get a much higher penalty than with the GDL alone (the converse being true for FNs). This is illustrated in Figure 5, where small roads are recovered, a demonstration of BL's effectiveness at mediating the precision/recall tradeoff in road segmentation problems, and an illustration of it's possibilities on the highly diverse and unbalanced data that satellite images of roads can be.

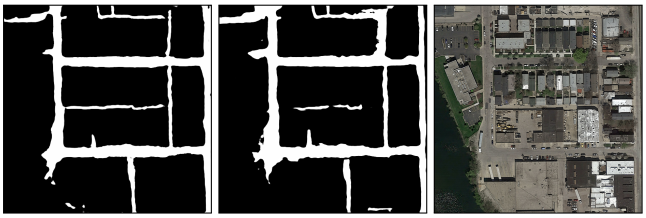

Figure 5: Combined loss mask (α = 0.005), Dice loss mask and original satellite image.

5 Ethical Risk Analysis

The most concerning ethical risk that we identified in our project is the possibility to retrace social and economics inequalities among diverse groups of people. This risk significantly impacts residents of less developed or maintained areas, where roads are more likely to be smaller, less visible, or surrounded by clutter. This negative aspect might translate into a bias in the development of infrastructure[16], favouring already well-developed areas with better road visibility. The risk is considerable in terms of severity, as it could widen the socio-economic gap.

Since the risk we identified is primarily related to the likelihood of misclassifications, and thus to the proportion of false positive and false negative predictions, we use the F1 scores of our validation and test data to quantify the risk. For validation we obtain an F1 score of approximately 0.90 with our different custom-tailored loss functions. These steps allowed us to better understand the model's performance and, therefore, the potential real-world impact.

To face the ethical risk associated with our segmentation task, we focused on enhancing the model's overall accuracy, delving into the critical balance between precision and recall. Our approach involved not only data augmentation, but also the implementation of custom-tailored loss functions specifically designed to optimize this balance. By prioritizing both Dice loss and Boundary loss in our training process, we focused on achieving an accurate segmentation that precisely delineates road edges and maintains pixel-wise accuracy. This approach allowed us to address the considerable challenge represented by the predominance of non-road regions by diminishing the risk of misclassification.

However, we acknowledge that the effectiveness of this approach is yet to be fully examined as our original dataset size is fairly small and lacks variability. A more inclusive dataset, particularly enriched with images from underdeveloped areas, would be crucial to assess the efficacy of our approach.

References

[1] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations”, *arXiv preprint arXiv:1707.03237*, 2017.

[2] H. Kervadec, J. Bouchtiba, C. Desrosiers, *et al.*, “Boundary loss for highly unbalanced segmentation”, in *Proceedings of the Machine Learning Research*, M. Niethammer, M. Styner, S. Aylward, *et al.*, Eds., vol. 102, London, United Kingdom: PMLR, 13–15 Apr 2019, pp. 285–302. [Online]. Available: https://proceedings.mlr.press/v102/kervadec19a.html.

[3] J. Jordan, *An overview of semantic image segmentation*, https://www.jeremyjordan.me/semantic-segmentation/, 2018.

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, *arXiv preprint arXiv:1505.04597*, 2015.

[5] 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.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[7] M. Huh, P. Agrawal, and A. A. Efros, “What makes imagenet good for transfer learning?”, *arXiv preprint arXiv:1608.08614*, 2016.

[8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

[9] Segmentation Models Contributors, *Segmentation models python library*, Year. [Online]. Available: URL%20of%20the%20library.

[10] W. Falcon and The PyTorch Lightning team, *PyTorch Lightning*, version 1.4, Mar. 2019. DOI: 10.5281/zenodo.3828935. [Online]. Available: https://github.com/Lightning-AI/lightning.

[11] SciPy Developers, *Scipy: Open source scientific tools for python*, Year. [Online]. Available: https://www.scipy.org.

[12] Author, “A methodology for evaluating image segmentation algorithms”, *Journal Name*, Year, Description of the usage of Intersection over Union in the context of image segmentation or object detection.

[13] S. Wang, G. Sun, B. Zheng, and Y. Du, “A crop image segmentation and extraction algorithm based on mask rcnn”, *Entropy*, vol. 23, no. 9, p. 1160, 2021. DOI: 10.3390/e23091160. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34573785/.

[14] M. S. Aloun, M. S. Hitam, W. N. J. H. W. Yussof, and Z. Bachok, “A review paper on image segmentation techniques based on colour and texture features”, *AIP Conference Proceedings*, vol. 2484, no. 1, p. 060 013, 2023. DOI: 10.1063/5.0114074. [Online]. Available: https://pubs.aip.org/aip/acp/article/2484/1/060013/2879587/A-review-paper-on-image-segmentation-techniques.

[15] O. Team, *Morphological transformations*, Accessed: December 21, 2023, 2023. [Online]. Available: https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html.

[16] Y. Zhu, L. Long, J. Wang, J. Yan, and X. Wang, “Road segmentation from high-fidelity remote sensing images using a context information capture network”, *Cognitive computation*, pp. 1–14, 2022.