ROADWHERE: A UNet model for road detection through semantic segmentation of road images

Vijay Jaisankar, Jaya Sreevalsan Nair

(a) Traffic scenario

(b) Binary road mask

Fig. 1: An illustration of the binary road mask for a traffic scenario.

***Abstract*— In this paper, we propose ROADWHERE, a UNet model for road detection. We explore dataset size reduction through difference hashing and evaluate the efficacy of pre-trained CNN encoders in semantic segmentation. By using a pre-trained VGG-19 backbone, we obtain a validation IOU score of 70.02% using just 12.20% of the training data.**

I. INTRODUCTION

A. Automated Driving Systems

The quality and robustness of Automated Driving Systems(ADSs) have greatly improved in the era of Deep Learning (DL). They are increasingly adopted to realise their potential benefits like preventing accidents, reducing emissions, transporting the mobility-impaired and reducing driving related stress [1]. The McKinsey Center for Future Mobility Analysis shows that autonomous driving could generate $300 Million to $400 Million by 2035 [2].

B. Semantic Segmentation

Semantic segmentation refers to the pixel-wise labeling of an image [3]. As opposed to conventional object detection, the pixel-wise fine-grained outputs of semantic segmentation models provide additional rigour to ADSs.

C. Road Detection

In this paper, we consider the task of **Road detection**. In particular, given an RGB image of a traffic scenario, our task is to localise the pixels containing the road surface.

As shown in Figure 1, our aim is to differentiate the pixels of the input image belonging to the road (shown in white) from the rest of the scenario (shown in black). This is an important basic task for autonomous vehicles attempting to navigate real-world environments.

II. DATASETS

A. Description of Candidates

There are multiple labelled datasets for the task of Automotive Semantic Segmentation. In this work, we consider the following candidates:

- CityScapes [4] contains $\approx$ 3000 images for training. The original dataset consists of 30 classes, of which we limit our work to locating the pixels classified as *flat:road*.
- KITTI-Road-Segmentation [5] is an aggregation of sources like the KITTI Vision Benchmark [6]. It contains $\approx$ 250 images for training.

B. The SizeDiv Metric

For efficient and powerful training, a candidate dataset should have sufficient number of samples and diversity across them. In this regard, we propose the following metric to assess the quality of the dataset candidate.

1) Notation:

- N - the number of scenario images in the training dataset
- u_i - the embedding vector for the image indexed i $(1 \leq i \leq N)$.

2) Methodology: As alluded to in II-B, our candidate dataset should have (1) adequate samples and (2) high diversity $\rightarrow$ low collective similarity between samples. To compute similarity between samples, we use a pre-trained BLIP [7] feature extractor, as hosted on LAVIS [8]. For an image indexed i, BLIP produces u_i, a 768-dimensional vector, which can be compared with other images' vectors for similarity.

For a given dataset with N images in its training dataset, we define $SizeDiv(N)$ as

$$SizeDiv(N) = N \cdot \frac{\binom{N}{2}}{\sum_{i=1}^{i=N-1} \sum_{j=i+1}^{j=N} u_i \cdot u_j}$$

Which equates to N scaled to the inverse of a joint similarity score of the images as computed through their embeddings.

TABLE I: SizeDiv scores of candidate datasets

Dataset name	N	SizeDiv score ($x10^6$)
CityScapes	2975	**0.676**
KITTI-Road-Segmentation	250	0.060

Based on our results summarised in Table I, we note that CityScapes not only provides more data points but is also more diverse, hence leading to a better SizeDiv score. Hence, we perform our experiments on the CityScapes dataset.

III. Training set: Size reduction

As shown in I, the Cityscapes dataset has vast diversity within its images. As we are working with a sub-problem of the original 30-class semantic segmentation task, we hypothesise that we can achieve decent results using a *subset* of the allocated training set. In this regard, we explore the possibility of the *Difference Hashing* algorithm as a technique to remove duplicates, resulting in a minimal and diverse dataset for training.

Hamming distance is used as a proxy for the *distance* between images. Two hashes with a Hamming distance of zero implies that the two hashes are identical (since there are no differing bits) and that the two images are identical/perceptually similar as well [9]. We use the maximum hamming distance between images as a fast threshold to remove duplicates.

TABLE II: Reduced dataset size after Difference Hashing

Maximum Hamming Distance	#Images in reduced training dataset
10	2852
15	1762
20	363
25	51
30	10

IV. UNet

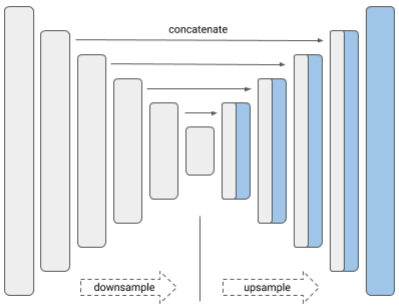

Fig. 2: An overview of the UNet Architecture.

The UNet [10] is one of the most legendary architectures in the field of semantic segmentation. As shown in Figure 2, the UNet consists of two symmetrical blocks - the *Encoder* and the *Decoder*.
The encoder successively down-samples the images through Convolutional layers into a *bottleneck* vector, which is then successively up-sampled by the decoder.
Crucially, *skip connections* between the symmetrical stages of the encoder and decoder ensure that the model learns a mixture of low-level features and high-level features. In UNet, these features are concatenated. In LinkNet [11], a variant of the UNet architecture, these features are added.

Building on the merits of transfer learning, pre-trained Convolutional Neural Networks (CNNs) are often used as backbones for the Encoder blocks. For our use-case, we make use of their immense representative power and utilise CNNs trained on Imagenet [12] as the encoders.

V. Building on representations

1) Choosing an Encoder: We employ the following models as candidates for encoders through preliminary experiments:

- *resnet-18* from the ResNet family of models [13] was chosen due to its smaller footprint.
- *inception-v3* from the Inception family of models [14] was chosen due its deep architecture in lieu of number of layers.
- *vgg-19* from the VGG family of the models [15] was chosen due to its large depth and number of parameters.

To reduce the overall training time, we keep the encoders' weights frozen and split the training section of Cityscapes into a *Train* section for this experiment (80%) and *Validation* section for this experiment (20%).

TABLE III: Training and Validation IOU Scores for the Encoder Selection Experiment

Architecture	Encoder	Train IOU	Val IOU
UNet	resnet-18	53.1772	52.9185
UNet	inception-v3	53.2342	52.9186
UNet	vgg-19	**56.1889**	**53.9431**
LinkNet	resnet-18	53.0129	52.9185
LinkNet	inception-v3	53.4080	52.9186

Through the results summarised in Table III, we choose the vanilla *UNet* architecture with *vgg-19* as the backbone.

2) ROADWHERE: With this background information, we set the layers of the vgg-19 encoder to trainable and run the model with the Adam optimiser [16] for 200 epochs.
We also apply the following augmentations to the training data at random:

- Changing the brightness of the images (75% original brightness $\leq$ augmented brightness $\leq$ 125% original brightness)
- Changing the contrast of the images (75% original contrast value $\leq$ augmented contrast value $\leq$ 125% original contrast value)

Our proposed solution, *ROADWHERE*, was trained on the reduced Cityscapes dataset filtered with a maximum hamming distance of 20 (see Table II). The validation dataset for this experiment is the entire Validation section of the Cityscapes dataset.

Over the course of its training loop, it achieved the following IOU scores:

TABLE IV: Training and Validation IOU Scores of ROADWHERE

Metric	Value
Training IOU Score	75.2172
Validation IOU Score	70.0203

VI. Discussions

A. Synthetic Data Generation

Recent advances in diffusion and energy-based models have led to a boom in iterative image editing model pipelines like DDPM Inversion [17]. We hypothesise that the results

(a) Base image given to the LEDITS model

(b) LEDITS' addition of a pothole

(c) LEDITS' addition of a pedestrian

(d) LEDITS' addition of rain to the scene

Fig. 3: Qualitative results of the synthetic data generation experiment.

of the "stricter" settings for Difference Hashing (i.e., higher values of the maximum hamming distance threshold) are good candidates for adding artefacts in order to generate synthetic traffic scenarios.

In this regard, we run the Difference Hashing experiment on the Validation section of the Cityscapes dataset setting the threshold to 30.

We make use of the publicly-available Gradio space of the LEDITS project [18] to attempt to generate the following artefacts:

- Potholes - this was chosen as a representative for physical objects
- Rain - this was chosen as a representative for simulating adversarial weather conditions
- Pedestrians

Qualitative results of this experiment are shown in Figure 3. We can see significant hallucination through addition of additional artefacts of the model in Figure 3d, wherein the trees on the right side of the frame has been replaced by a building. We also note that the content guidance factor must be increased to avoid cases like that of Figure 3c wherein the desired artefacts are not added to the image at all. From our experiments, we conclude that addition of relative smaller physical objects like potholes and traffic cones is most feasible for this approach.

B. Future Work

We note the following themes for future work for ROADWHERE.

- Experimenting with different semantic segmentation architectures like Lightweight CNN backbone-based, Multibranch backbone-based, and Transformer-based archirectures [19].
- Adversarial training and evaluating the robustness of ROADWHERE in the face of adversarial attacks like [20].

VII. Conclusion

In this paper, we have looked at the formulation of ROADWHERE, a VGG-19 UNet model for detecting road pixels in an image through semantic segmentation. We have also looked at the benefits of effective dataset size reduction through the results obtained. We also note some interesting future directions for this project.

VIII. Acknowledgement

We would like to acknowledge the following Github repositories for their valuable implementations of crucial sections of ROADWHERE:

- https://github.com/qubvel/segmentation_models
- https://github.com/idealo/imagededup
- https://github.com/salesforce/LAVIS
- https://github.com/camenduru/ledits-hf

References

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[2] J. Deichmann, E. Ebel, K. Heineke, R. Heuss, M. Kellner, and F. Steiner, "Autonomous driving's future: Convenient and connected," Jan 2023. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected

[3] I. Ulku and E. Akagündüz, "A survey on deep learning-based architectures for semantic segmentation on 2d images," *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2032924, 2022. [Online]. Available: https://doi.org/10.1080/08839514.2022.2032924

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] S. Mahna, "Kitti-road-segmentation," Oct 2021. [Online]. Available: https://www.kaggle.com/datasets/sakshaymahna/kittiroadsegmentation/

[6] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[7] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, "BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation," *CoRR*, vol. abs/2201.12086, 2022. [Online]. Available: https://arxiv.org/abs/2201.12086

[8] D. Li, J. Li, H. Le, G. Wang, S. Savarese, and S. C. Hoi, "LAVIS: A one-stop library for language-vision intelligence," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 31–41. [Online]. Available: https://aclanthology.org/2023.acl-demo.3

[9] A. Rosebrock, "Image hashing with opencv and python," Apr 2021. [Online]. Available: https://pyimagesearch.com/2017/11/27/image-hashing-opencv-python/

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[11] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4.

[12] 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*, 2009, pp. 248–255.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '16. IEEE, Jun. 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[17] I. Huberman-Spiegelglas, V. Kulikov, and T. Michaeli, "An edit friendly ddpm noise space: Inversion and manipulations," *arXiv preprint arXiv:2304.06140*, 2023.

[18] L. Tsaban and A. Passos, "Ledits: Real image editing with ddpm inversion and semantic guidance," 2023.

[19] J. Cheng, H. Li, D. Li, S. Hua, and V. S. Sheng, "A survey on image semantic segmentation using deep learning techniques," *Computers, Materials amp; Continua*, vol. 74, no. 1, p. 1941–1957, 2023. [Online]. Available: http://dx.doi.org/10.32604/cmc.2023.032757

[20] G. Lim, M. Kim, and J. Hur, "Adversarial attack on semantic segmentation preprocessed with super resolution," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, pp. 484–490.