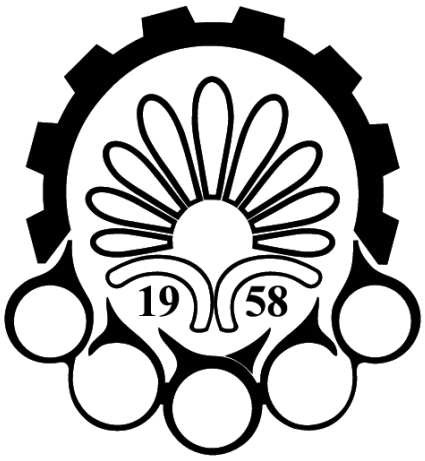


Amirkabir University of Technology
(Tehran Polytechnic)

Binary semantic segmentation for asphalt cracks report

Amirhossein Ghadiri

Feb 2024

1 Problem Statement

Asphalt crack segmentation is essential for maintaining infrastructure integrity and safety. It enables early detection and repair of cracks, preventing further damage and hazards on roads and pavements. Timely maintenance not only ensures road safety but also saves costs by avoiding more extensive repairs or reconstructions. Additionally, it contributes to smoother traffic flow, reduces environmental impact, and helps in collecting valuable data for future maintenance planning (Baduge et al., 2023; Chun et al., 2021; Feng et al., 2020). Therefore, this study aims to develop an asphalt crack segmentation model based on PyTorch. The best performance is sought and the model is evaluated based on two metrics: Dice and Jaccard (IoU) (Taha and Hanbury, 2015).

2 Methodology and results

The environment used for training and benchmarking the model was Google Colab. The GPU that was assigned was a Tesla T4 with 12.7 GB of RAM and 78.2 GB of disk available. Initially, MAnet (Fan et al., 2020) was selected as the architecture for the model as it has performed well on the dataset of the MICCAI 2017 LiTS (Bilic et al., 2023) Challenge. Regarding the encoder, mit_b5 (Xie et al., 2021) was exploited which combines an efficient mechanism with strong performance. Dice loss is a famous and widely used loss function for image segmentation tasks (Jadon, 2020) that was chosen as the loss function here. Previous studies have suggested batch sizes of 32 and 64 for training segmentation models (Perdomo et al., 2017). In this study, however, even a batch size of 16 resulted in memory issues; consequently, 4 and 8 were selected as the batch sizes. Finally, Adam and RMSprop were set as the optimizers (Ruder, 2017).

The training phase began with Learning Rate=0.0001. If no reduction was recorded in validation Dice Loss for 5 consecutive epochs, the learning rate was multiplied by a factor of 0.2. Also, considering that the Dice metric was not implemented in the original repository, it was implemented manually by the author. Albumentations was used for data augmentation. The training was followed based on the abovementioned settings for 75 epochs. Figure 1 shows the result of the training on the validation dataset for the best model in each configuration.

	Architecture	Encoder	Optimizer	Batch Size	IoU	Dice	FPS
1	MAnet	mit_b5	Adam	8	0.6496	0.7669	13
2	MAnet	mit_b5	Adam	4	0.6397	0.7599	13
3	MAnet	mit_b5	RMSprop	8	0.6427	0.7616	13
4	MAnet	mit_b5	RMSprop	4	0.6384	0.7589	13

Figure 1: Training outputs.

Figure 1 shows that the best optimizer and batch size combination for this task is setting Adam as the optimizer and 8 as the batch size. Based on this finding regarding the training settings, another architecture with two other encoders where also trained. In particular, FPN (Lin et al., 2017) was selected as the alternative architecture and efficientnet-b0 and efficientnet-b7 (Tan and Le, 2019) where the alternative encoders. The validation result of the training for the previous best and the alternative models are shown in Figure 2.

	Architecture	Encoder	Optimizer	Batch Size	IoU	Dice	FPS
1	MAnet	mit_b5	Adam	8	0.6496	0.7669	13
2	FPN	efficientnet-b0	Adam	8	0.6293	0.7518	28
3	FPN	efficientnet-b7	Adam	8	0.6418	0.7622	14
4	FPN	efficientnet-b0	Adam	8	0.6187	0.7437	31
5	FPN	efficientnet-b7	Adam	8	0.6314	0.7537	15

Figure 2: Training outputs for the previous best and the alternative models.

As Figure 2 demonstrates, all models operate in real-time (5 FPS is conventionally selected as the margin for real-time.) Considering that selecting MAnet as the architecture and mit_b5 as the encoder has yield the best performance, the first model is selected as the base model of this study. The superiority of this model is most probably because both MAnet and mit_b5 are more state-of-the-art in comparison to the alternative models. The base model recorded IoU=0.5789 and Dice=0.7181 on the test dataset. Figure 3 shows the training curve for training the base model.

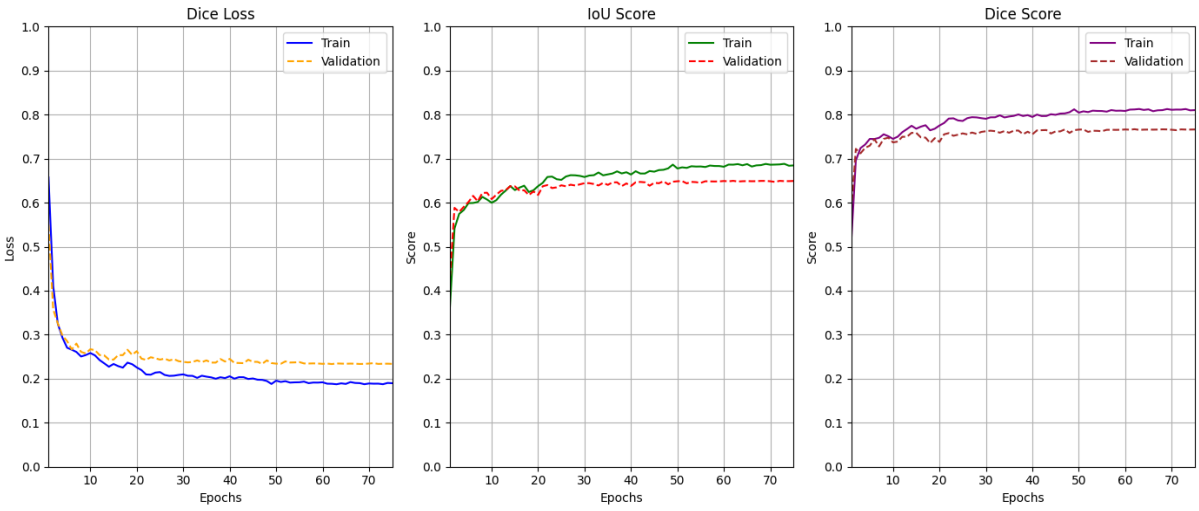
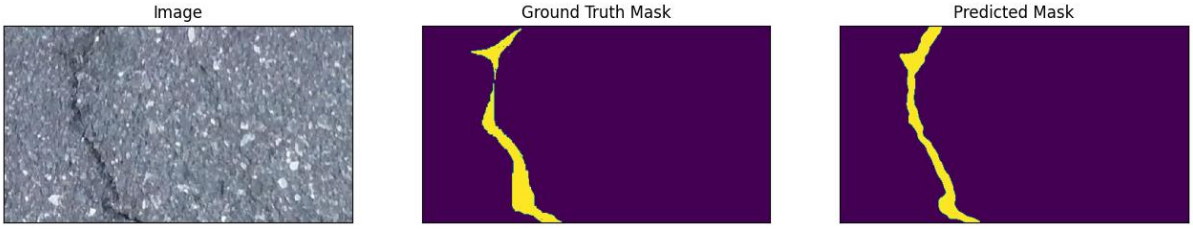


Figure 3: Training curve for the base model.

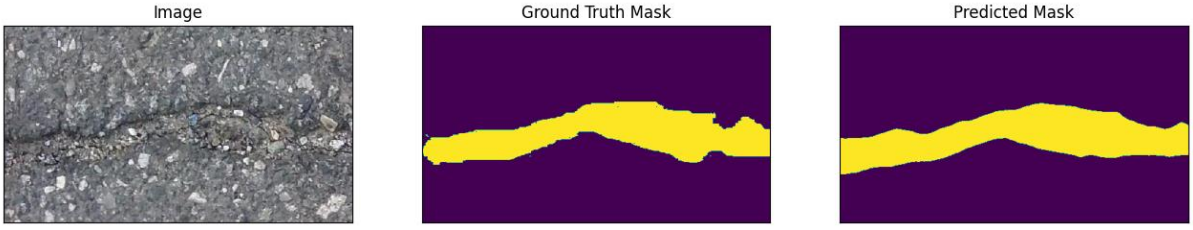
3 Discussion

Figure 4 compares the outputs of the base model with the ground truth on some of the test images. Although, the evaluation metrics on the test dataset may allude that the final model is not practical, comparing the predicted masks with the ground truth masks visually makes it clear that the model indeed segments the asphalt cracks effectively. There are a few possible explanations for why the evaluation metrics are not high. Firstly, the area of the cracks in each image contains a low amount of its area as the cracks are narrow. Thus, a tiny error in segmenting the crack results in a significantly lower intersection of the predicted and ground truth masks. Secondly, the ground truth masks are not extremely accurate. For instance, it is

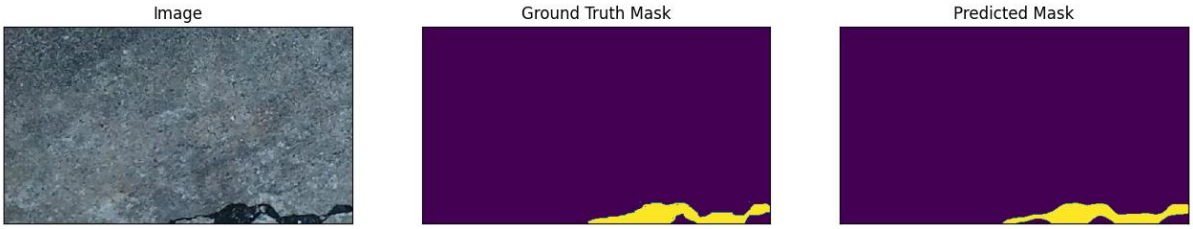
obvious that in Figure 4(c), Figure 4(m), and Figure 4(u), the predicted masks are more accurate than the ground truth masks. In contrast to labeling other objects such as humans and cars for image segmentation, labeling asphalt cracks is rather arbitrary, and every individual may label the images differently.



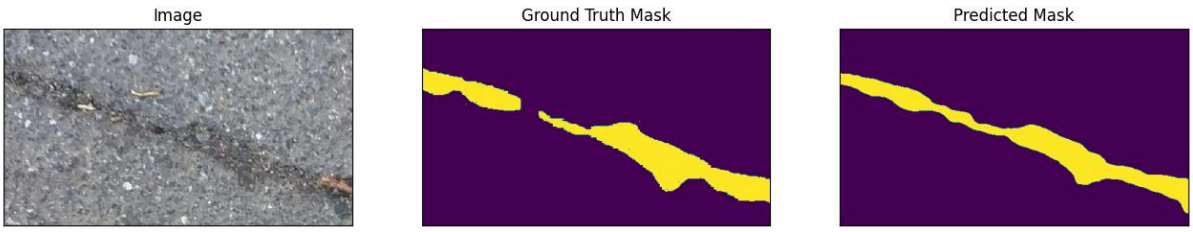
(a)



(b)



(c)



(d)

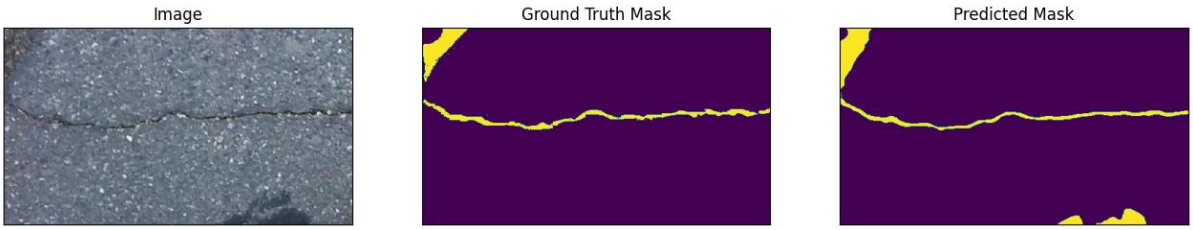


Image
Ground Truth Mask
Predicted Mask

(e)

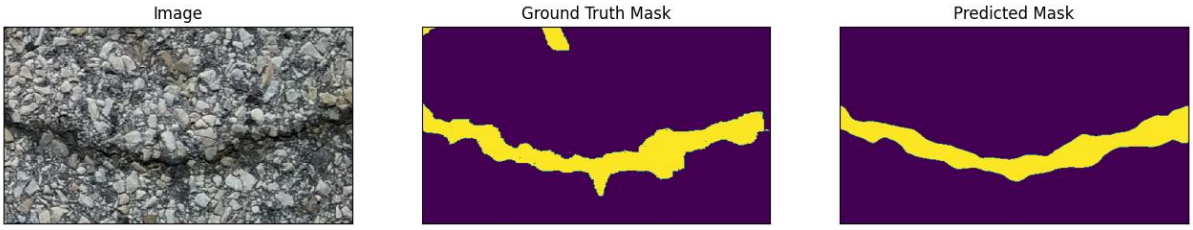


Image
Ground Truth Mask
Predicted Mask

(f)


Image
Ground Truth Mask
Predicted Mask

(g)

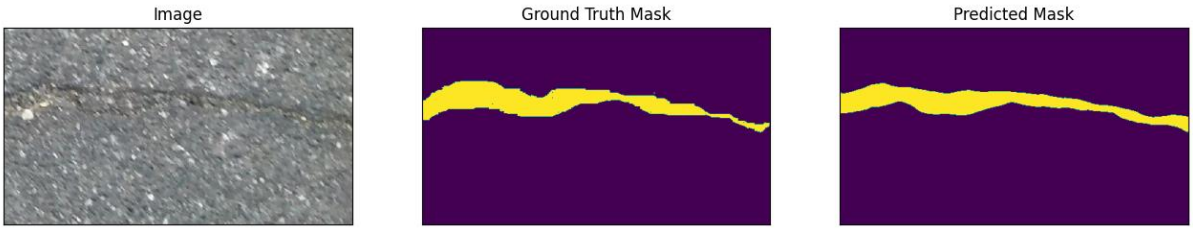


Image
Ground Truth Mask
Predicted Mask

(h)


Image
Ground Truth Mask
Predicted Mask

(i)

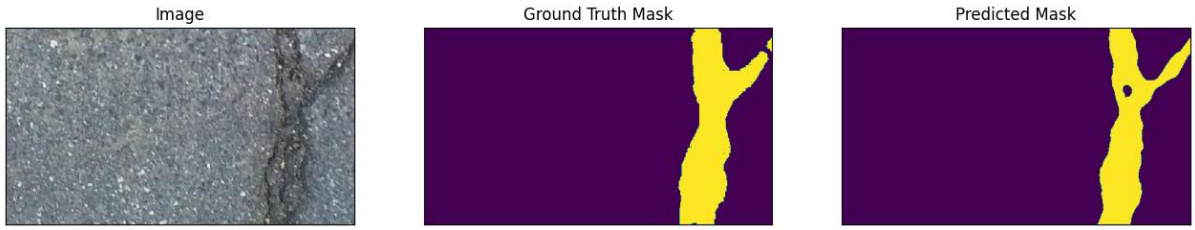


Image
Ground Truth Mask
Predicted Mask

(j)

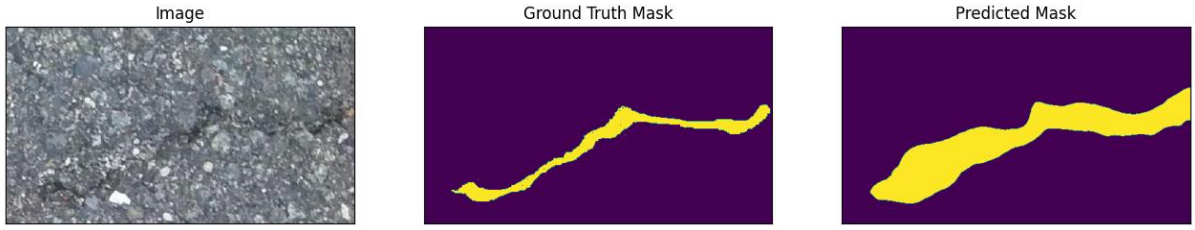


Image
Ground Truth Mask
Predicted Mask

(k)

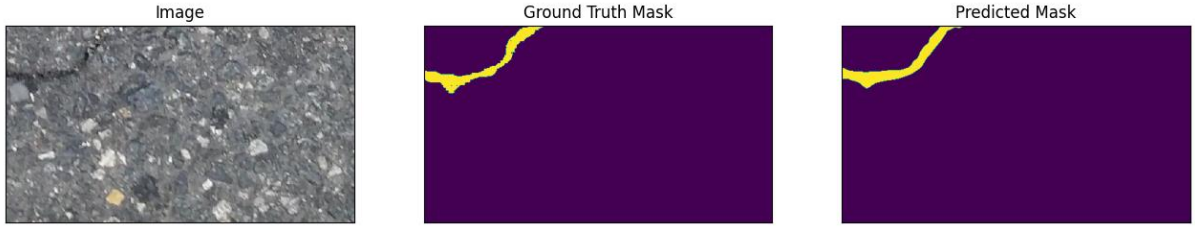


Image
Ground Truth Mask
Predicted Mask

(l)

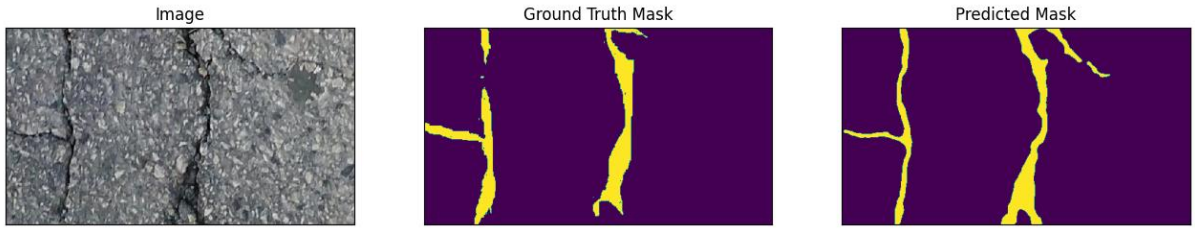


Image
Ground Truth Mask
Predicted Mask

(m)

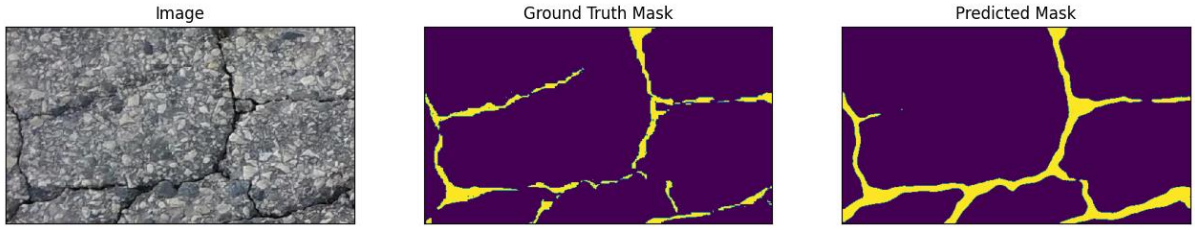


Image
Ground Truth Mask
Predicted Mask

(n)

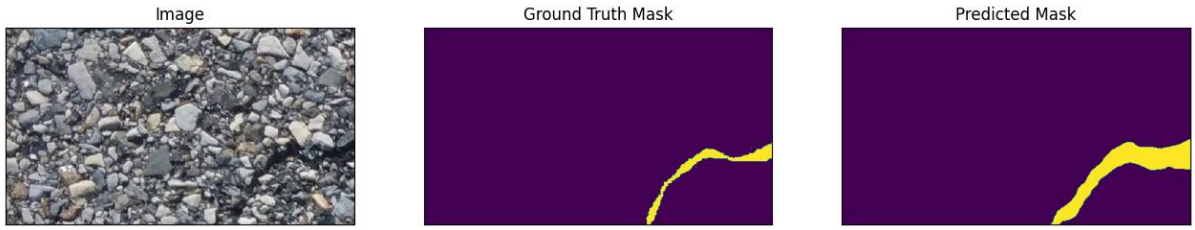


Image
Ground Truth Mask
Predicted Mask

(o)


Image
Ground Truth Mask
Predicted Mask

(p)


Image
Ground Truth Mask
Predicted Mask

(q)

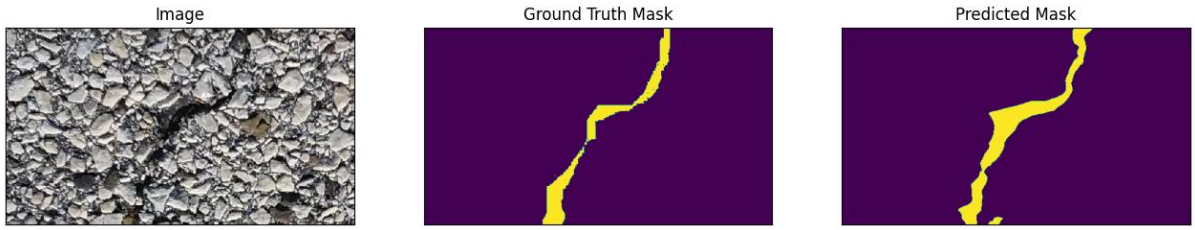


Image
Ground Truth Mask
Predicted Mask

(r)

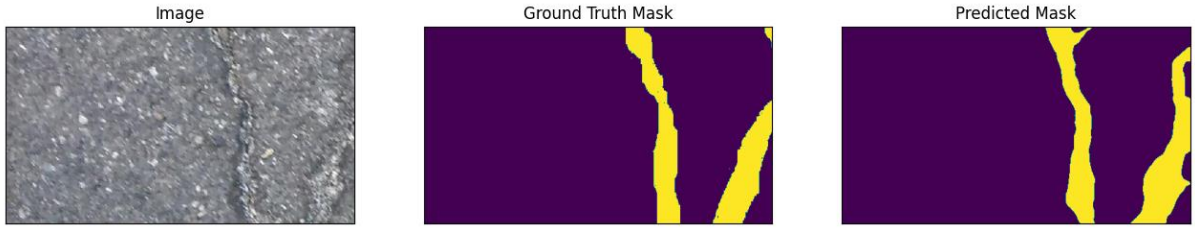


Image
Ground Truth Mask
Predicted Mask

(s)

(t)

(u)

Figure 4: model outputs compared with ground truths.

References

Baduge, S.K., Thilakarathna, S., Perera, J.S., Ruwanpathirana, G.P., Doyle, L., Duckett, M., Lee, J., Saenda, J., Mendis, P., 2023. Assessment of crack severity of asphalt pavements using deep learning algorithms and geospatial system. Constr Build Mater 401. https://doi.org/10.1016/j.conbuildmat.2023.132684

Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G.E.H., Chartrand, G., Lohöfer, F., Holch, J.W., Sommer, W., Hofmann, F., Hostettler, A., Lev-Cohain, N., Drozdzal, M., Amitai, M.M., Vivanti, R., Sosna, J., Ezhov, I., Sekuboyina, A., Navarro, F., Kofler, F., Paetzold, J.C., Shit, S., Hu, X., Lipková, J., Rempfler, M., Piraud, M., Kirschke, J., Wiestler, B., Zhang, Z., Hülsemeyer, C., Beetz, M., Ettlinger, F., Antonelli, M., Bae, W., Bellver, M., Bi, L., Chen, H., Chlebus, G., Dam, E.B., Dou, Q., Fu, C.W., Georgescu, B., Giró-i-Nieto, X., Gruen, F., Han, X., Heng, P.A., Hesser, J., Moltz, J.H., Igel, C., Isensee, F., Jäger, P., Jia, F., Kaluva, K.C., Khened, M., Kim, I., Kim, J.H., Kim, S., Kohl, S., Konopczynski, T., Kori, A., Krishnamurthi, G., Li, F., Li, H., Li, J., Li, X., Lowengrub, J., Ma, J., Maier-Hein, K., Maninis, K.K., Meine, H., Merhof, D., Pai, A., Perslev, M., Petersen, J., Pont-Tuset, J., Qi, J., Qi, X., Rippel, O., Roth, K., Sarasua, I., Schenk, A., Shen, Z., Torres, J., Wachinger, C., Wang, C., Weninger, L., Wu, J., Xu, D., Yang, X., Yu, S.C.H., Yuan, Y., Yue, M., Zhang, L., Cardoso, J., Bakas, S., Braren, R., Heinemann, V., Pal, C., Tang, A., Kadoury, S., Soler, L., van Ginneken, B., Greenspan, H., Joskowicz, L., Menze, B., 2023. The Liver Tumor Segmentation Benchmark (LiTS). Med Image Anal 84. https://doi.org/10.1016/j.media.2022.102680

Chun, P.J., Yamane, T., Tsuzuki, Y., 2021. Automatic detection of cracks in asphalt pavement using deep learning to overcome weaknesses in images and gis visualization. Applied Sciences (Switzerland) 11. https://doi.org/10.3390/app11030892

Fan, T., Wang, G., Li, Y., Wang, H., 2020. Ma-net: A multi-scale attention network for liver and tumor segmentation. IEEE Access 8. https://doi.org/10.1109/ACCESS.2020.3025372

Feng, X., Xiao, L., Li, W., Pei, L., Sun, Z., Ma, Z., Shen, H., Ju, H., 2020. Pavement Crack Detection and Segmentation Method Based on Improved Deep Learning Fusion Model. Math Probl Eng 2020. https://doi.org/10.1155/2020/8515213

Jadon, S., 2020. A survey of loss functions for semantic segmentation, in: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020. https://doi.org/10.1109/CIBCB48159.2020.9277638

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. https://doi.org/10.1109/CVPR.2017.106

Perdomo, O., Arevalo, J., González, F.A., 2017. Convolutional network to detect exudates in eye fundus images of diabetic subjects, in: 12th International Symposium on Medical Information Processing and Analysis. https://doi.org/10.1117/12.2256939

Ruder, S., 2017. An Overview Optimization Gradients. arXiv preprint arXiv:1609.04747.

Taha, A.A., Hanbury, A., 2015. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. BMC Med Imaging 15. https://doi.org/10.1186/s12880-015-0068-x

Tan, M., Le, Q. V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks, in: 36th International Conference on Machine Learning, ICML 2019. https://doi.org/10.48550/arXiv.1905.11946

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, in: Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.2105.15203