Mapping of Coastline and Drone Object Detection

AI Research Intern at Ranvox Technology Pvt. Ltd.

Kovid Sharma
School of Mechanical Sciences
Indian Institute of Technology Bhubaneswar
26 September 2024

Table of Contents

Introduction

Unmanned Aerial Vehicles (UAVs) and drones have gained extensive acceptance in both academic study and practical applications[1]. Drones are extensively utilised and implemented in various fields, such as agriculture production[2], disaster relief[3], Wildlife Protection[4] and Urban Surveillance[5]. Utilising computer vision in drone-captured scenarios is a recently trendy endeavour. It is one of the crucial technologies that will enhance the perceptive abilities of drones.

The problem description provided consisted of two tasks. First and foremost, to assess the present state of Object Detection in scenarios captured by drones and its real-time implementation on Edge devices. Furthermore, to develop a model that accurately represents the coastline of an ocean or water body.

Object detection is a popular subject in UAV applications, ensuring the accurate iden tification and categorisation of all things captured in UAV images, including pedestrians, vehicles, bicycles, and others. Deep learning-based object detection has shown substantial advancement in both accuracy and efficiency.

The coastline is an important feature of the coastal morphology. Monitoring coastal zones is crucial in safeguarding the environment, while identifying coastlines is essential for efficiently managing coastal areas[6]. Various factors impact the Coastline, including erosion, sedimentation, human activities and climate change[7]. Such repercussions justify the surveillance of coastlines. For our task, Semantic Segmentation was chosen as the approach for coastline detection.

Fig. 1. Images at different timestamps from a video. It shows that even if the coast profile is same it is difficult to identify due to the occlusion and camera angle

A collection of beach photographs captured by UAVs at different angles was generated by getting stock videos from the internet. Afterwards, each video was divided into several images and their respective masks were created. Fig. 1 illustrates that each shot in the video had a similar coast profile, but it was difficult to observe due to the occlusion caused by the ocean waves. U-Net Architecture[8] was used to segment the coastline. Coastline detection was necessary to support the object detection model in determining whether an object was located in the sea or on the beach.

The methodology for coastline segmentation used U-Net as the model. The model was pre-trained on the Binary Tree Segment Dataset[9] since the new beach dataset only had 230 images. Therefore, it was important to use data augmentation while training.

Related Work

The current deep learning-based object detectors are divided into two broad categories: one-stage and two-stage detectors. Typically, the two-stage[10-12] approach attains a high level of accuracy but lacks efficiency, making it

to execute on a UAV platform with restricted computational resources. They employ a region-proposal network to ascertain whether the previous anchors correspond to an object or background. The previous anchors consist of multiple explicitly defined potential bounding boxes. Then, they employ two head networks to categorise potential anchors and estimate the offset between the anchors and ground truth boxes. One-stage detectors[13-14] eliminate the region proposal network completely. The categories and offsets of the past anchors are predicted directly using two detectors. A novel category of detectors, anchor-free detectors[15-16], has been recently introduced. Their approach simplifies the bounding box prediction to the key point and size estimation. It presents an improved method for detecting objects with various scales.

The direct application of prior models to address object detection tasks in drone-captured scenarios is not feasible for three primary reasons[17]. Firstly, due to the significant fluctuations in flight altitude of drones, the scale of the objects undergoes vigorous variations. Furthermore, drone-captured photos may include very high-density objects, resulting in occlusion. Drone-captured photos inherently include confusing geographical features due to their coverage of large regions. Other than the above reasons, the restricted computational resources of edge GPU devices make real-world object detection on UAVs a difficult problem.

[17] used an additional prediction head to detect different-scale objects. They replaced the original heads with transformer prediction heads to introduce self-attention mechanism. They integrated the convolutional block attention module(CBAM) to find attention regions in scenarios with dense objects. Performed experimentation on the VisDrone 21 dataset and achieved state-of-the-art results.

[18] mixed-up anchor-free detectors with a re-regression module. The detector first generates the coarse boxes and then applies the re-regression module to generate the accurate bounding boxes.

[19] proposed a novel lightweight architecture based on the YOLOX model for real-time object detection on Edge GPUs like Jetson NX and Jetson Nano. They designed an efficient and lightweight PixSF head, which included an attention module. A slimmer neck structure, SlimFPN, was developed to reduce the network parameters. They achieved a better trade-off between accuracy and latency on the VisDrone21[] dataset.

[20] employed an aerial image of the region sourced from Google Maps to detect the coastline. However, coastline mapping from UAVs using segmentation algorithms is still yet to be explored.

Dataset and model description

Dataset Details

The dataset was created by sourcing stock videos from the website DroneStock[21]. Videos were converted to images using OpenCV, `cv2.VideoCapture` method was used to encode the video and convert it to frames. All the videos were 30 FPS and had a resolution of 1920×1080. Images were taken every 30 ms and were resized to a resolution of 512×512 using `cv2.Resize` method. The images were labelled using the semantic segmentation workspace of Label Studio[22]. Binary masks created were exported in PNG format. Fig. 3 shows an example of an image and its respective mask. The final dataset contains 230 image-mask pairs.

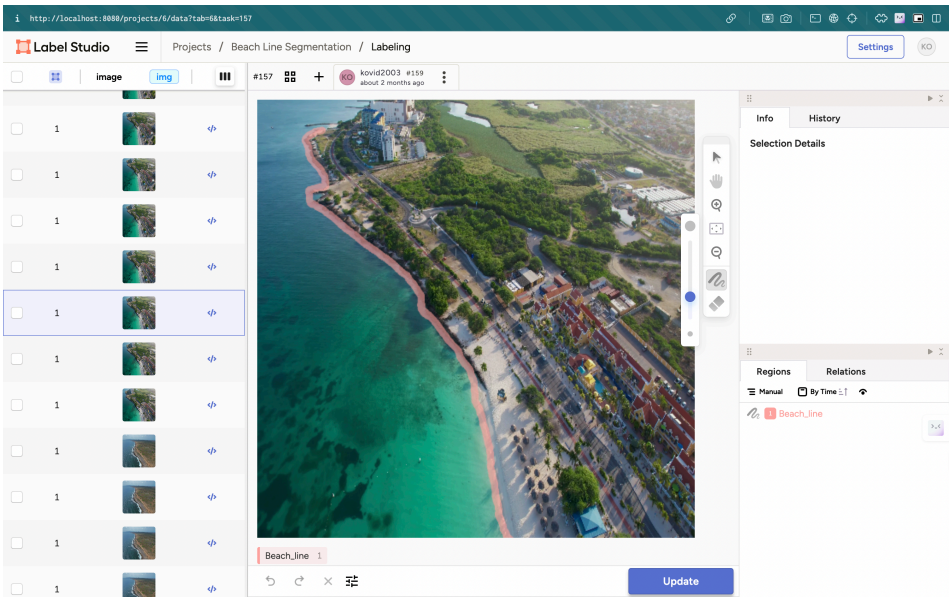

Fig. 2: Semantic Segmentation workspace

Fig. 3: Image and its respective mask

Model Details

The model used for the segmentation task is U-Net[8]. Fig. 4 shows the structure of the model. Python and PyTorch were used to create the model. For the down-sampling layer, convolutional layers were created using `torch.nn.Conv2d` along with ReLU activation(`torch.nn.ReLU`). Batch normalisation(`torch.nn.BatchNorm2d`) was used for faster convergence and, therefore, faster training. It is also required since we have a huge class imbalance.

Max pooling was used to reduce the resolution(`torch.nn.MaxPool2d`). Similarly, Transpose convolutional layers were created for the up-sampling layer using `ConvTranspose2d`. The model contains 31, 037, 633 trainable parameters.

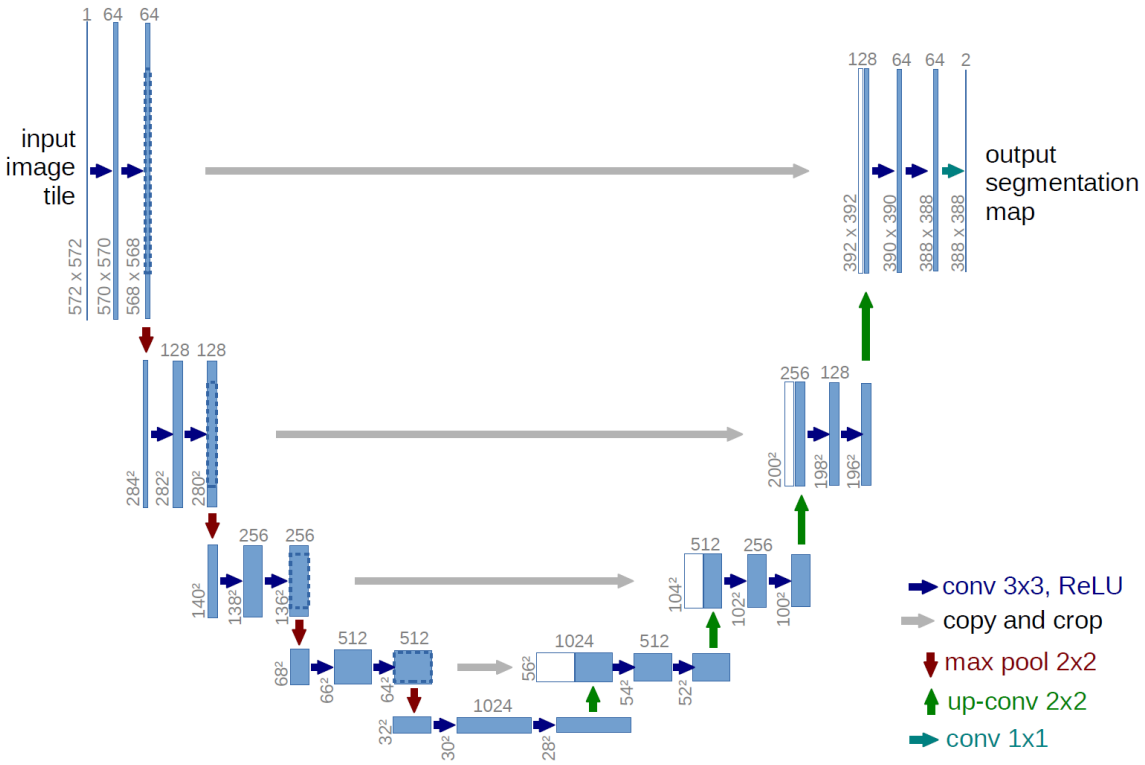

Fig. 4 U-Net Architecture[]

Experimentation Details

The model was trained on an NVIDIA P100 GPU provided by Kaggle, containing 16 GB of GPU memory. The model was pre-trained on the Binary Tree Segmentation dataset[9]. The loss function used is Weighted Binary cross-entropy with logits loss. This loss combines a Sigmoid layer and the BCELoss in one single class. This version is more numerically stable than using a plain Sigmoid followed by a BCELoss by combining the operations into one layer[23].

The optimiser used for training is the Adam optimiser [24, 25]. Data augmentation like Rotation, HorizontalFlip, and Vertical Flip were applied using the albumentations library[26].

For training, Automatic Mixed Precision is used to improve training performance and reduce memory usage. It was implemented using `torch.amp` library[]. The model was pre-trained for 30 Epochs and trained on the main dataset for 50 Epochs. The learning rate schedule for training is shown in Fig. 5a and 5b.

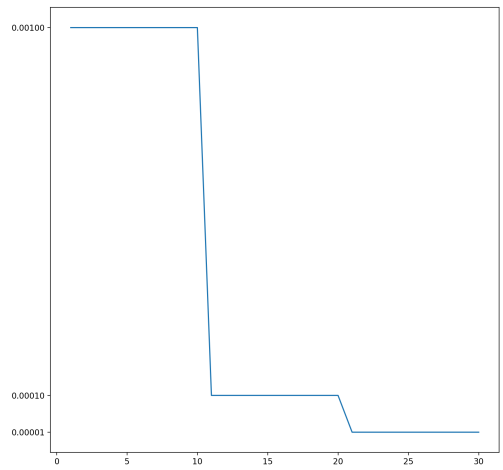

Fig. 5a: Learning Rate Schedule for Pre-training

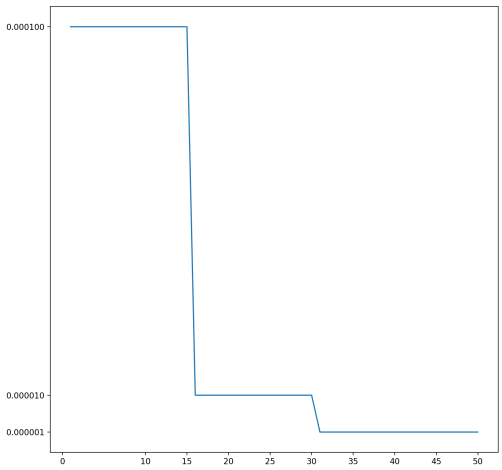

Fig. 5b: Learning Rate Schedule for training

Results and Discussions

The model achieved 90.0394% accuracy and a 0.223 dice score, which is a decent score since our dataset only has 230 images. But it still needs improvement as a dice score greater than 0.8 is considered a successful segmentation. Accuracy is a misleading metric for this task, as the true values in the binary mask are significantly less than the false values. Therefore, it is important to consider only the dice score as a metric. The scores can be further improved by pre-training on a large dataset. Fig. 6 shows the variation in accuracy and dice score.

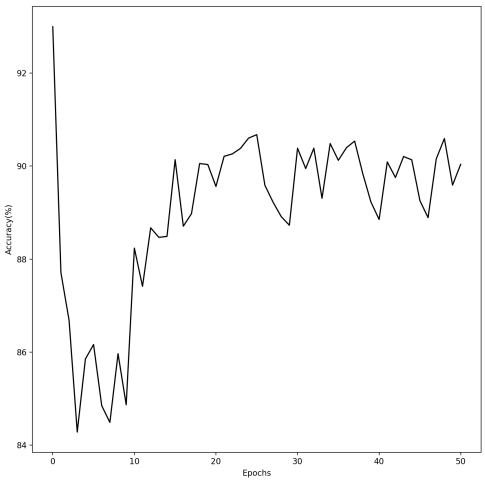

Fig. 6a: Accuracy Vs Epochs

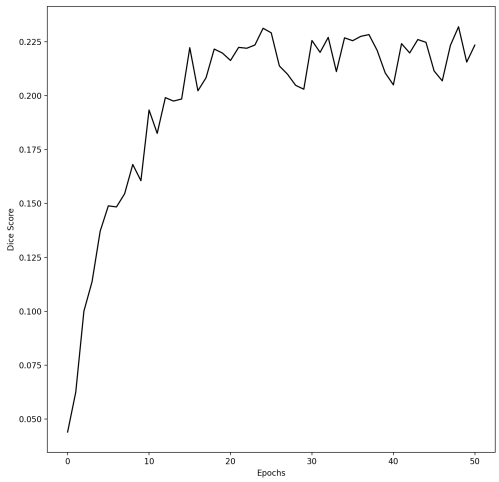

Fig. 6a: Dice Score Vs Epochs

Conclusion and Future Prospects

The segmentation performed using U-Net was successful, and segmentation models can be used for coastline detection. Further experimentation on U-Net can be done by creating a larger dataset. Modern segmentation models like SegFormer[28] can be used to detect the shoreline. Segmentation features learned can be used to detect obstacles by creating a detection head and training the model for detection tasks. A completely separate model can also be prepared for object detection. A new dataset for Indian scenarios can be created to cater to Indian coastal requirements.

References

1. Zhu, Pengfei & Wen, Longyin & Mamgain, Nehal & Vedurupaka, Naveen & Joseph, K & Balasubramanian, Vineeth. (2018). VisDrone-DET2018: The Vision Meets Drone Object Detection in Image Challenge Results.
2. Zhenfeng Shao, Congmin Li, Deren Li, Orhan Altan, Lei Zhang, and Lin Ding. An accurate matching method for projecting vector data into surveillance video to monitor and protect cultivated land. ISPRS Int. J. Geo Inf., 9(7):448, 2020.
3. A. Shamsoshoara, F. Afghah, A. Razi, S. Mousavi, J. Ashdown and K. Turk, "An Autonomous Spectrum Management Scheme for Unmanned Aerial Vehicle Networks in Disaster Relief Operations," in IEEE Access, vol. 8, pp. 58064-58079, 2020, doi: 10.1109/ACCESS.2020.2982932.
4. Benjamin Kellenberger, Michele Volpi, and Devis Tuia. Fast animal detection in UAV images using convolutional neural networks. In 2017 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2017, Fort Worth, TX, USA, July 23-28, 2017, pages 866–869. IEEE, 2017.
5. Jingjing Gu, Tao Su, Qiuhong Wang, Xiaojiang Du, and Mohsen Guizani. Multiple moving targets surveillance based on a cooperative network for multi-uav. IEEE Commun. Mag., 56(4):82–89, 2018.
6. Kolednik, D. Coastal monitoring for change detection using multi-temporal LiDAR data. In Proceedings of the CESCG 2014: The 18th Central European Seminar on Computer Graphics, Smolenice, Slovakia, 25–27 May 2014; pp. 73–78.
7. Roshanka Ranasinghe, Assessing climate change impacts on open sandy coasts: A review, Earth-Science Reviews, Volume 160, 2016, Pages 320-332, ISSN 0012-8252, https://doi.org/10.1016/j.earscirev.2016.07.011.
8. Olaf Ronneberger, , Philipp Fischer, Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." (2015).
9. https://www.kaggle.com/datasets/earthshot/tree-binary-segmentation
10. R. Girshick, “Fast r-cnn,” in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.
11. S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137– 1149, 2017.

12. T.-Y.Lin, P.Dolla´r, R.Girshick, K.He, B.Hariharan, and S.Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

13. Joseph Redmon, Santosh Divvala, Ross Girshick, & Ali Farhadi. (2016). You Only Look Once: Unified, Real-Time Object Detection.

14. Liu, Wei & Anguelov, Dragomir & Erhan, Dumitru & Szegedy, Christian & Reed, Scott & Fu, Cheng-Yang & Berg, Alexander. (2016). SSD: Single Shot MultiBox Detector. 9905. 21-37. 10.1007/978-3-319-46448-0_2.

15. Zhi Tian et al., "FCOS: Fully Convolutional One-Stage Object Detection," 2019.

16. Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, & Jian Sun. (2021). YOLOX: Exceeding YOLO Series in 2021.

17. X. Zhu, S. Lyu, X. Wang and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 2021, pp. 2778-2788, doi: 10.1109/ICCVW54120.2021.00312.

18. C. Chen et al., “RRNet: A Hybrid Detector for Object Detection in Drone-Captured Images,” in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 100–108. doi: 10.1109/ICCVW.2019.00018.

19. Zhou, W., Min, X., Hu, R., Long, Y., Luo, H., & JunYi. (2022). FasterX: Real-Time Object Detection Based on Edge GPUs for UAV Applications. https://arxiv.org/abs/2209.03157

20. Heidarsson, Hordur & Sukhatme, Gaurav. (2011). Obstacle detection from overhead imagery using self-supervised learning for Autonomous Surface Vehicles. IEEE International Conference on Intelligent Robots and Systems. 3160-3165. 10.1109/IROS.2011.6048233.

21. https://dronestock.com/

22. https://labelstud.io/

23. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

24. Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980

25. https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

26. https://github.com/albumentations-team/albumentations

27. https://pytorch.org/docs/stable/amp.html

28. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. https://arxiv.org/abs/2105.15203