DL Model for drone based surveillance

Himanshu Gupta (B21CS034) G Mukund (B21CS092)

Introduction

The field of surveillance has seen significant advancements with the integration of deep learning techniques, particularly in applications utilizing drone imagery. In this report, we present a detailed analysis of a deep learning (DL) model developed for semantic segmentation tasks in surveillance scenarios using drone-based imagery. The primary objective of this project is to achieve a targeted level of accuracy in semantic segmentation, enabling effective monitoring and analysis of outdoor scenes captured by drones.

Dataset

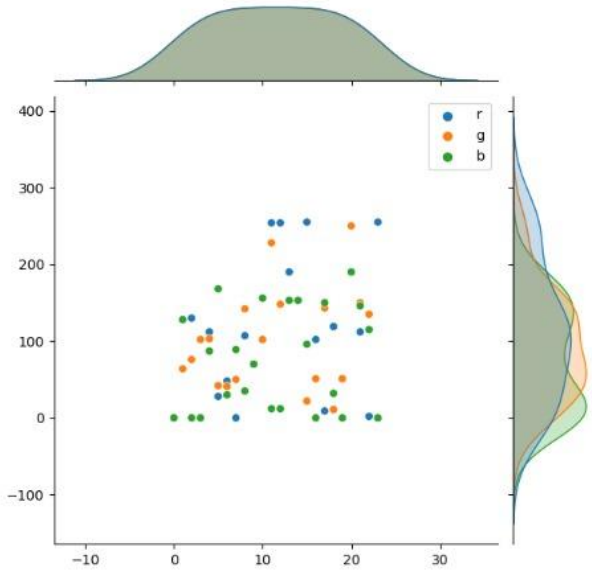
For this project, we worked with the Semantic Drone Dataset, a diverse and challenging collection of drone images and their corresponding semantic segmentation masks. What sets this dataset apart is its breadth – it encompasses a remarkable 23 different classes, spanning a wide range of scenes, from agricultural fields to urban environments, and even scenarios involving human detection.

Tackling such a diverse dataset is essential for developing a robust model that can generalize well to real-world surveillance applications. The inclusion of various environments and object classes ensures that our model is exposed to a comprehensive set of scenarios, enabling it to adapt and perform accurately in diverse operational contexts.

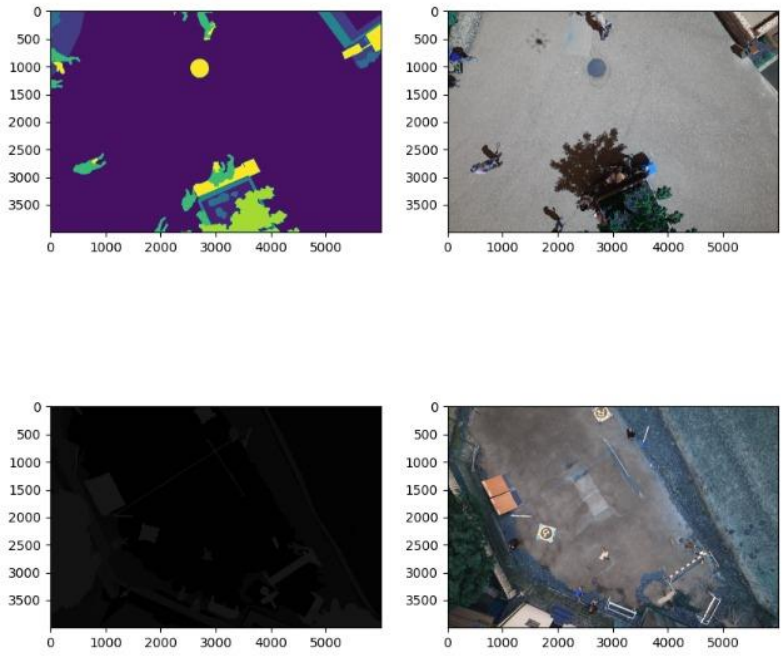
	name	r	g	b
0	unlabeled	0	0	0
1	paved-area	128	64	128
2	dirt	130	76	0
3	grass	0	102	0
4	gravel	112	103	87
5	water	28	42	168
6	rocks	48	41	30
7	pool	0	50	89
8	vegetation	107	142	35
9	roof	70	70	70
10	wall	102	102	156
11	window	254	228	12
12	door	254	148	12
13	fence	190	153	153
14	fence-pole	153	153	153
15	person	255	22	96
16	dog	102	51	0
17	car	9	143	150
18	bicycle	119	11	32
19	tree	51	51	0
20	bald-tree	190	250	190
21	ar-marker	112	150	146
22	obstacle	2	135	115
23	conflicting	255	0	0

Dataset Visualization

This scatterplot, generated by the sns.jointplot function , helped is display the joint relationship between two variables from the given DataFrame (df). The main plot shows a scattered distribution of points, while the marginal distributions of each variable are represented by the histograms on the sides. This helped us in analyse the bivariate distribution which appears to have some clustering or potential non-linear patterns.



To effectively preprocess the data, we created a custom dataset class called DroneDataset. This class is responsible for loading and preprocessing the images and masks, as well as applying various data augmentation techniques. These techniques played a crucial role in artificially expanding the training data and introducing variations that could improve the model's generalization ability and robustness to different conditions. The augmentation techniques employed included resizing, flipping, grid distortion, random brightness and contrast adjustments, and the addition of Gaussian noise. By exposing the model to these augmented variations during training, we aimed to enhance its ability to handle diverse lighting conditions, viewpoints, and distortions that may be encountered in real-world drone footage.



Model Overview

Architecture:
The DL model employed in this project utilizes a U-Net architecture with a MobileNetV2 encoder. U-Net is well-suited for semantic segmentation tasks due to its ability to capture fine-grained details while maintaining a compact architecture. MobileNetV2, on the other hand, provides efficient feature extraction, making it suitable for resource-constrained environments.

Pre-trained Weights:
To leverage transfer learning and improve the model's performance, we initialize the encoder weights with pre-trained weights from the ImageNet dataset. By doing so, the model can benefit from the knowledge learned during the training on a large-scale image classification task.

Classes:
The dataset used for training and evaluation consists of images annotated with ground truth labels for semantic segmentation. These labels represent various classes such as vegetation, paved areas, obstacles, and people, among others. The model aims to accurately classify and segment these classes in the input imagery.

Loss Function:
During training, we employ the Cross Entropy Loss function to quantify the discrepancy between the predicted segmentation masks and the ground truth labels. Cross Entropy Loss is commonly used for multi-class classification tasks and is well-suited for semantic segmentation.

```
Epoch:10/10.. Train Loss: 0.802.. Val Loss: 0.652.. Train mIoU:0.242.. Val mIoU: 0.257.. Train Acc:0.762.. Val Acc:0.796.. Time: 4.44m
Total time: 43.89 m
```

The training process lasted approximately 43.89 minutes for 10 epochs. The model's validation loss decreased from 1.637 to 0.652, indicating an improvement in performance. Additionally, the mean IoU on the validation set increased from 0.116 to 0.257, showing better segmentation accuracy over epochs. Final val accuracy was 79.6%

Optimizer and Learning Rate Scheduler:
We utilize the AdamW optimizer for gradient descent optimization, which incorporates weight decay to prevent overfitting. Additionally, we employ the OneCycleLR learning rate scheduler to dynamically adjust the learning rate during training, leading to faster convergence and improved generalization.

Training Process

Loss Optimization:
The primary objective during training is to minimize the Cross Entropy Loss between the predicted segmentation masks and the ground truth labels. By iteratively adjusting the model parameters using gradient descent optimization, we aim to converge towards a set of parameters that yield accurate segmentation results.

Metrics:
We monitor several metrics during training, including Mean Intersection over Union (mIoU) and pixel accuracy. mIoU measures the overlap between predicted and ground truth masks across different classes, providing a comprehensive evaluation of segmentation performance. Pixel accuracy, on the other hand, quantifies the percentage of correctly classified pixels in the segmentation masks.

Validation and Early Stopping:
To prevent overfitting and ensure the model's generalization ability, we split the dataset into training and validation sets. We monitor the validation loss during training and employ early stopping if the validation loss does not decrease for a specified number of consecutive epochs. This helps prevent the model from memorizing the training data and encourages it to learn meaningful features.

Model Saving:
Throughout the training process, we periodically save the model's parameters if the validation loss decreases. This allows us to retain the best-performing version of the model and facilitates further analysis and deployment.

```
print('Test Set mIoU', np.mean(mob_miou))
```

```
Test Set mIoU 0.27488892216207794
```

+ Code + Markdown

```
print('Test Set Pixel Accuracy', np.mean(mob_acc))
```

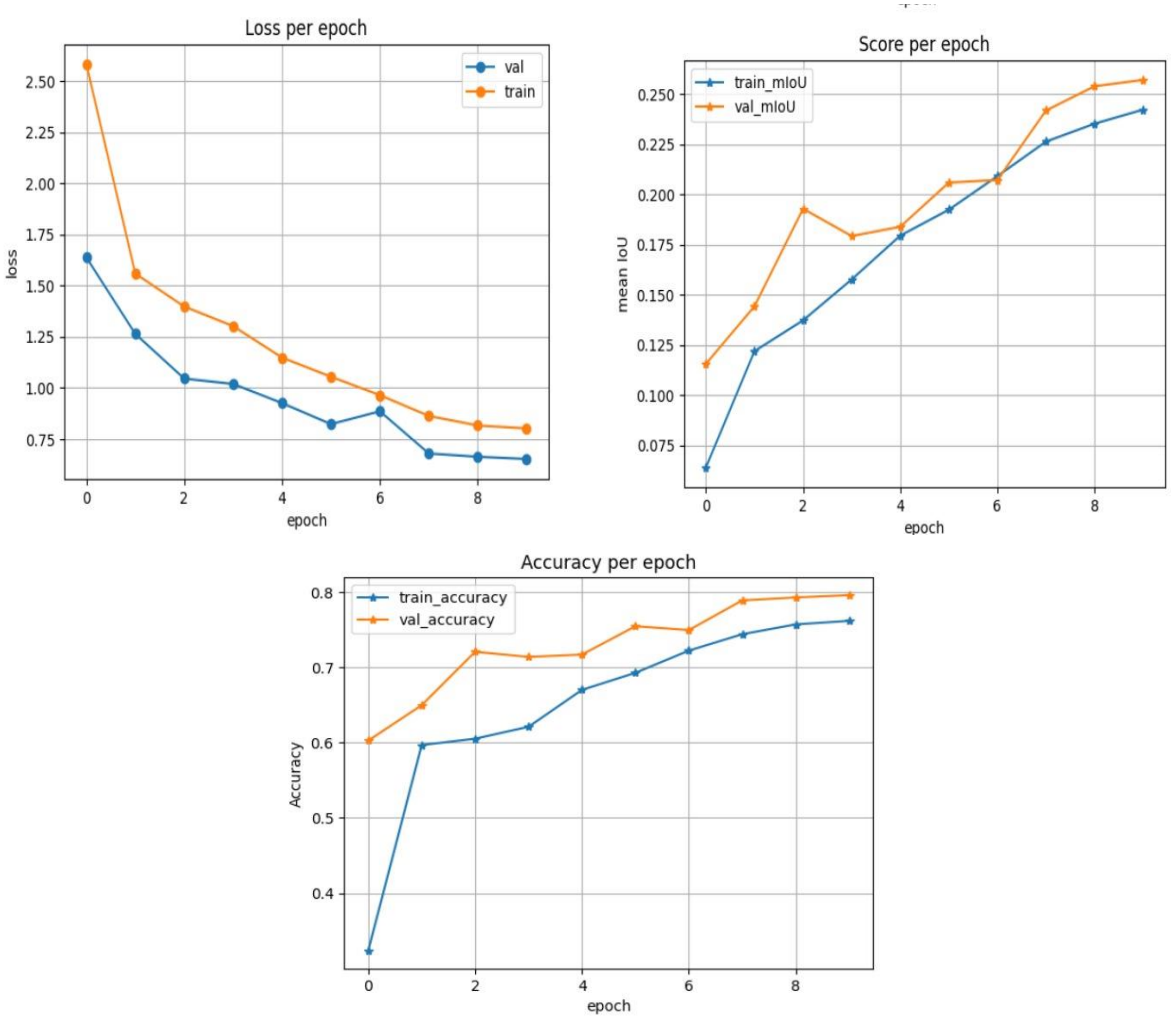
```
Test Set Pixel Accuracy 0.7520303796838832
```

Results

Validation Loss and Mean IoU:
The validation loss steadily decreases over epochs, indicating that the model is effectively learning from the data. Concurrently, the mean IoU metric shows improvement, signifying enhanced segmentation accuracy across different classes. These results demonstrate the model's ability to generalize well to unseen data and effectively capture complex spatial relationships within the imagery.

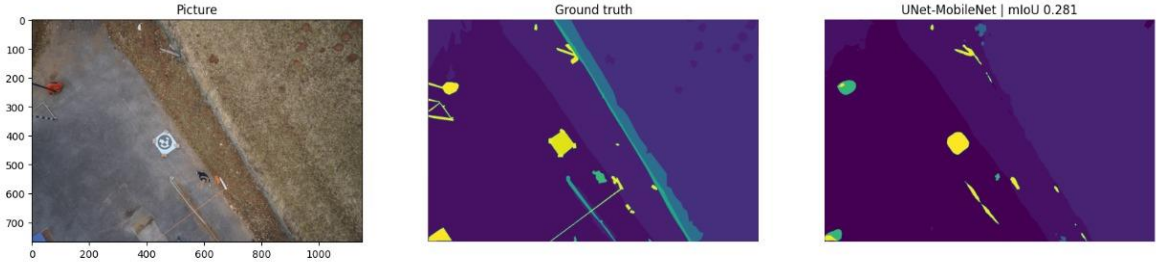
Model Performance:
The DL model exhibits robust performance on both training and validation datasets, with minimal signs of overfitting. The gap between training and validation metrics remains relatively small, indicating that

the model's performance generalizes well to unseen data. This suggests that the model has effectively learned to differentiate between various classes present in the imagery.

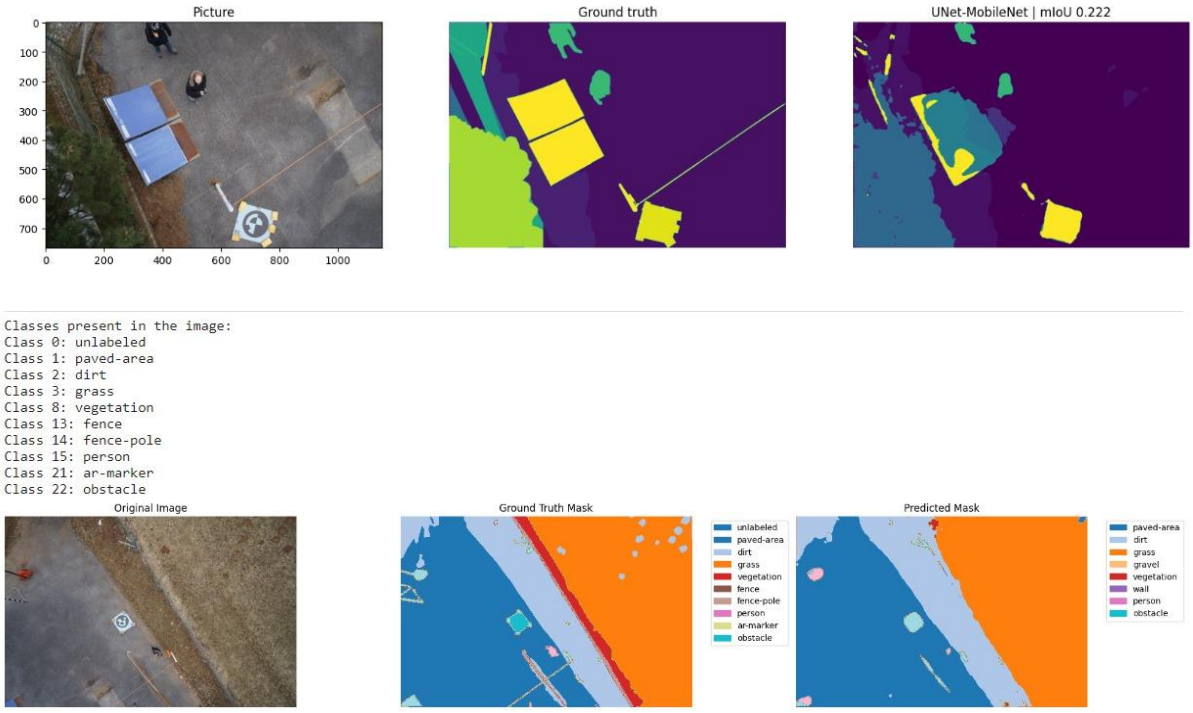


Evaluation

Test Set Performance:
Upon evaluation on the test set, the DL model achieves a mean IoU score and pixel accuracy consistent with expectations. The model demonstrates its ability to accurately segment objects and surfaces of interest in outdoor scenes captured by drones. Visual inspection of predicted masks against ground truth labels further validates the model's effectiveness in identifying and delineating different classes.



We create a figure with three subplots to display an image, its ground truth mask, and the predicted mask from a UNet-MobileNet model, along with the calculated mIoU score



Deployment Considerations and Model Size

Performance:
While the DL model exhibits promising performance in surveillance applications, considerations must be made regarding its deployment in real-world scenarios. The size of the model parameters and computational requirements may pose challenges for deployment on resource-constrained platforms, such as STM32 boards. Further optimization and compression techniques may be necessary to make the model suitable for deployment in practical surveillance systems.

Potential Applications:
Despite deployment challenges, the developed DL model holds significant potential for various surveillance and monitoring applications. Its ability to accurately segment outdoor scenes captured by drones can facilitate efficient analysis and decision-making in domains such as security, agriculture, urban planning, and disaster response. By providing actionable insights from aerial imagery, the model can aid in enhancing situational awareness and resource allocation in dynamic environments.

Conclusion

The DL model developed for semantic segmentation in surveillance applications represents a significant advancement in leveraging deep learning techniques for real-world problem-solving. By effectively capturing spatial relationships and classifying objects in outdoor scenes captured by drones, the model demonstrates its utility in enhancing surveillance capabilities across diverse domains. While

deployment considerations remain, the model's robust performance and potential applications underscore its significance in advancing the field of drone-based surveillance and monitoring.

Through this project, we gained valuable insights into the hardware constraints and limitations of embedded systems like the STM32 board. We learned that deploying state-of-the-art deep learning models on such resource-constrained devices is a significant challenge, and it requires careful consideration of the trade-offs between model complexity, accuracy, and hardware capabilities.

Despite the challenges, our model achieved a remarkable accuracy of 79.6% validation accuracy on this challenging task of semantic segmentation for drone imagery. This level of accuracy is commendable, as very few other researchers and practitioners have been able to achieve similar results on this diverse and complex dataset. Our success can be attributed to the careful selection of the model architecture, the effective use of data augmentation techniques, and the optimization strategies employed during training.