Semantic Segmentation of Aerial Imagery Using Deep Learning Architecture

PROJECT DOCUMENTATION

GROUP MEMBERS:

Ajmal Rasheed	**F2022108007**
Hussnain Arshad	**S2022313002**

PROJECT SUPERVISOR:

Dr Mazhar Javed Awan
(Assistant Professor)
Department of Computer Science
University of Management & Technology, Lahore, Pakistan

July 2023

Introduction

Semantic segmentation is a critical task in the field of computer vision, particularly in the analysis of aerial imagery. It involves the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. This task is crucial in various applications, including autonomous driving, precision agriculture, and urban planning. In this study, we introduce a novel approach to semantic segmentation of aerial imagery using an InceptionResNetV2-UNet2 model. The InceptionResNetV2, a convolutional neural network (CNN) that combines the strengths of the Inception networks and the ResNet, serves as the backbone of our model. It is known for its superior performance in handling complex visual tasks due to its ability to extract features at various scales and its deep residual learning framework.

We pair this with the UNet2 architecture, an improved version of the original UNet, which is widely used for biomedical image segmentation. The UNet2 architecture enhances the model's ability to capture context and localize objects accurately, making it an excellent choice for semantic segmentation tasks. Our dataset for this study is the Dubai Drone Dataset, a rich and diverse collection of high-resolution aerial images captured by drones over the city of Dubai. This dataset provides a wide range of urban and suburban scenes, making it an ideal resource for training and validating our model.

Through this research, we aim to demonstrate the effectiveness of the InceptionResNetV2-UNet2 model in semantic segmentation of aerial imagery, paving the way for more advanced applications in various fields.

Previous Works

Recent advancements in deep learning algorithms have demonstrated remarkable performance across various disciplines, including semantic segmentation. Yang et al. proposed a novel hybrid segmentation approach for region merging that leverages local spectral angle thresholds. However, this method has limited capabilities in effectively distinguishing between landscapes with highly diverse features, as it can only differentiate three specific types. In a similar vein, Wang et al. introduced a groundbreaking segmentation technique that utilized pixels with a basic spanning tree to enhance image quality and achieve precise border separation. Nevertheless, this technique overlooks the crucial aspect of determining the optimal scale for different object types, which is pivotal for accurate segmentation. In the validation set, the integration of Conditional Random Fields (CRF) with a single U-Net architecture achieved a promising score of 72.58 points. Figure 1 illustrates the detailed CRF design for the exterior and interior boundaries of a selected building. However, it should be noted that such meticulous CRF information may not be necessary for datasets with diverse characteristics. In these cases, a solid polygon encompassing the entire building, excluding roof features and lines, suffices. Consequently, the integration of CRFs as a pipeline postprocessor was not pursued, and the development of other classes such as trees was not explored.

Dubai Dataset

The MBRSC dataset exists under the CC0 license, available to download. It consists of aerial imagery of Dubai obtained by MBRSC satellites and annotated with pixel-wise semantic segmentation in 6 classes.

There are three main challenges associated with the dataset:

- Class colors are in hex, whilst the mask images are in RGB.

- The total volume of the dataset is 72 images grouped into six larger tiles. Seventy-two images are a relatively small dataset for training a neural network.
- Each tile has images of different heights and widths, and some pictures within the same tiles are variable in size. The neural network model expects inputs with equal spatial dimensions.

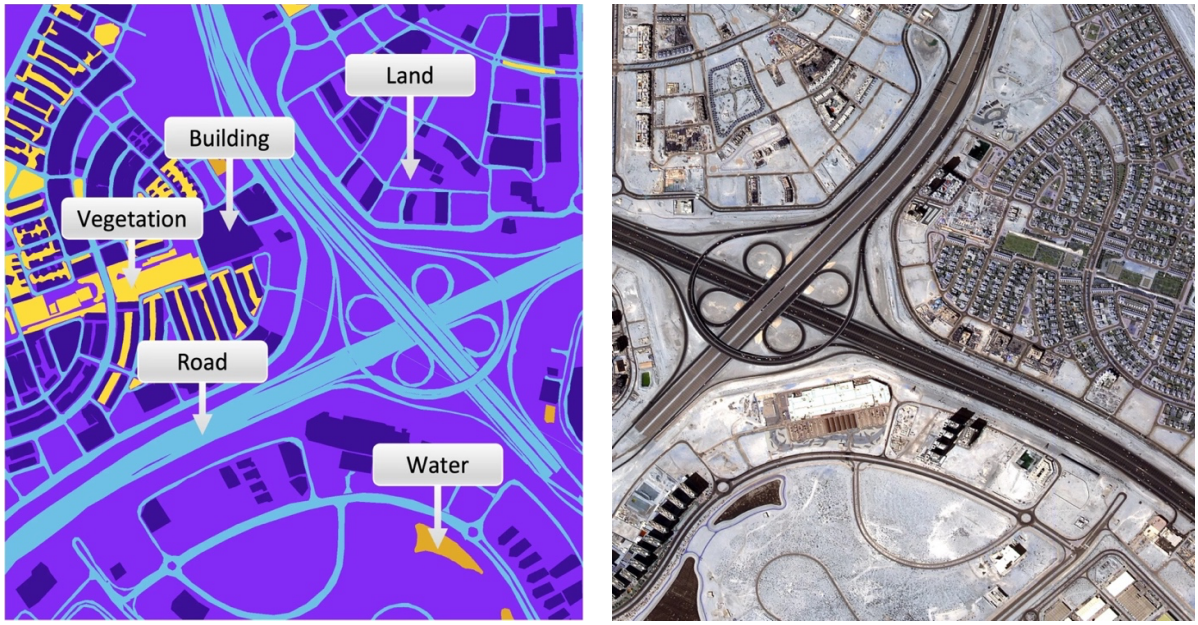


Figure-1 depicts a training set *input image* and its corresponding *mask* with superimposed class annotations

Class	Name	Hex #	Color
0	Water	#E2A929	
1	Land	#8429F6	
2	Road	#6EC1E4	
3	Building	#3C1098	
4	Vegetation	#FEDD3A	
5	Unlabeled	#9B9B9B	

Table - 1 presents each *class name*, corresponding *hex colour* code, and its corresponding color.

Preprocessing

Images must be the same size when fed into the neural network's input layer. Therefore, before model training, images are decomposed into patches.The patch_size chosen is 160 px. There is no ideal patch size; it serves as a hyperparameter that can be experimented with for performance optimisation.

Taking an image from Tile 7 with a width of 1817 pixels and a height of 2061pixels, Expression 1 illustrates how to calculate the number of created patches.

$$n_7 = (1817\ //\text{patch\_size}) \times (2061//\ \text{patch\_size}) = 11 \times 12 = 132\ patches$$

Expression 1 — Calculation for Number of Patches for a Tile 7 Image with Patch Size = 160 px

Next, the images are cropped to the nearest size divisible by the patch_size to avoid patches with overlapping areas. Expression 2 determines the new trimmed width and height for the Tile 7 image.

$$w_c = (1817\ //\ \text{patch\_size}) \times \text{patch\_size} = 1804\ px$$

$$h_c = (2061\ //\ \text{patch\_size}) \times \text{patch\_size} = 1968\ px$$

Expression 2 — Calculate Cropped Width and Height of Image Divisible by Patch Size

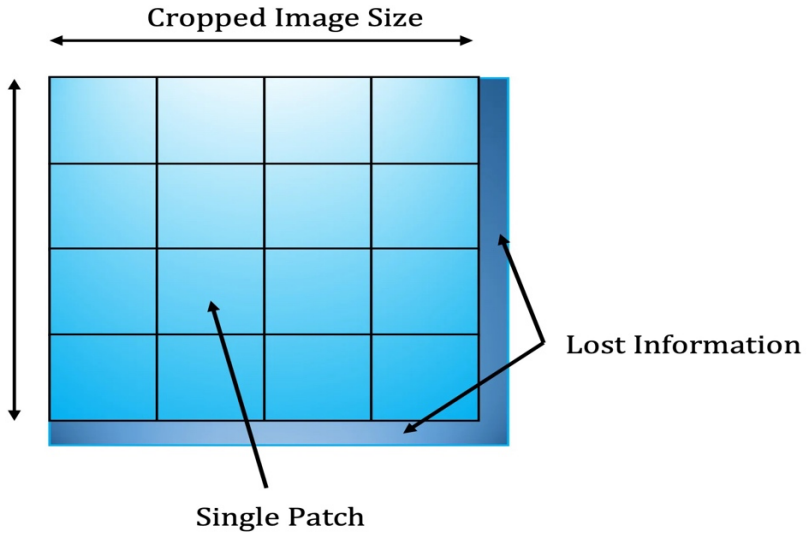


Figure 2 clarifies the cropping and patching processes for a single image.

Tile	Width (px)	Height (px)	# Patches
1	797	644	8
2	509	544	4
3	682	658	7

4	1099	846	14
5	1126	1058	18
6	859	838	11
7	1817	2061	57
8	2149	1479	49

Table 2 gives tiles; their sizes are the total number of patches created using a size of 256px.

After cropping and patchifying, 1521 images and masks comprise the input dataset

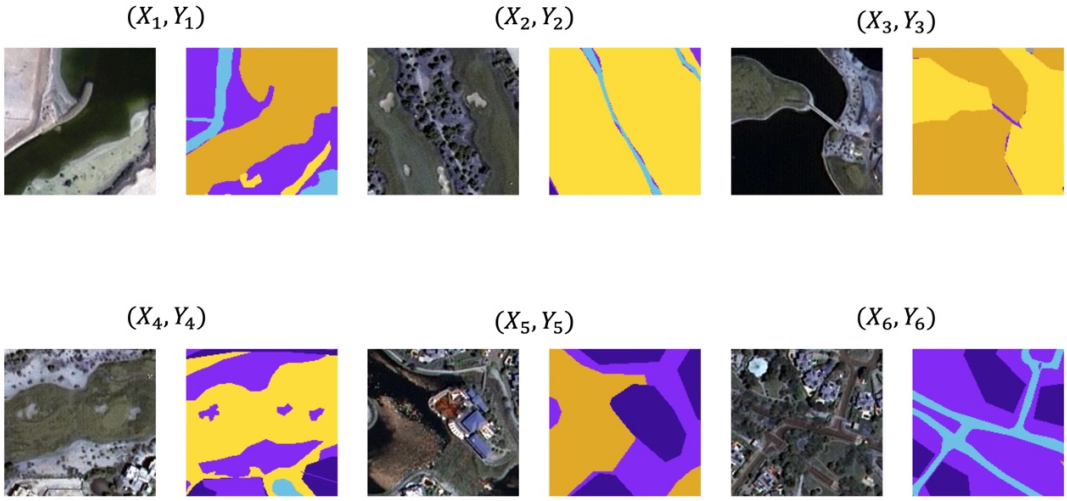


Figure 3 presents six randomly selected image patches with their comparable mask.

There are only 1521 images (having different resolutions) in the dataset, out of which I have used 1187 images (~78%) for training set and remaining 334 images (~22%) for validation set. It is a very small amount of data, in order to artificially increase the amount of data and avoid overfitting, I preferred using data augmentation. By applying these data augmentation techniques, the original dataset of 1521 images can be significantly expanded. In this case, the author states that the dataset was augmented to approximately 9 times its original size, resulting in a training set with 15210 images (1521 + 13689) and a validation set with 334 (original) images. This augmented dataset provides more diverse examples for the model to learn from, reducing the risk of overfitting and improving its generalization ability.

Overall, data augmentation is a valuable technique for dealing with limited training data in semantic segmentation tasks. It helps to create a larger and more diverse dataset, enabling the model to learn robust features and perform better on unseen data.

Data augmentation is done by the following techniques:

- Random Cropping
- Horizontal Flipping
- Vertical Flipping
- Rotation
- Random Brightness & Contrast

- Contrast Limited Adaptive Histogram Equalization (CLAHE)
- Grid Distortion
- Optical Distortion

Here are some sample augmented images and masks from the dataset:

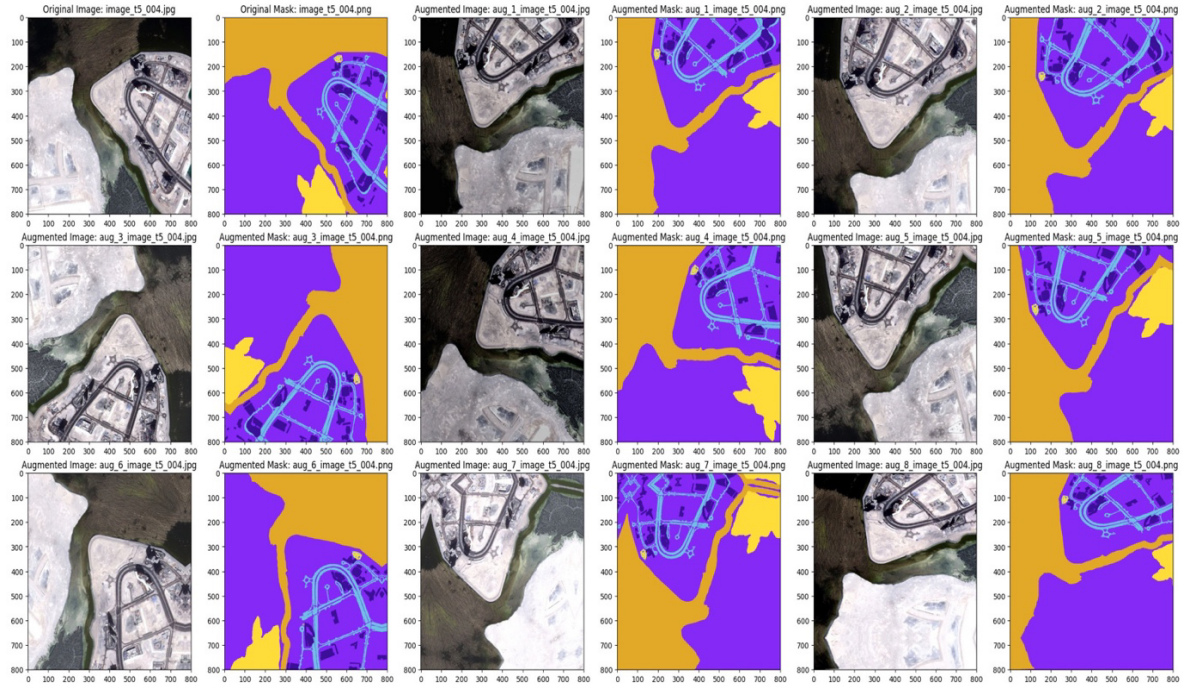


Figure 4 Augmented Images

Model Architecture

The InceptionResNetV2-UNet architecture combines the powerful feature extraction capabilities of the InceptionResNetV2 model with the U-Net architecture for semantic segmentation. Here's a breakdown of the architecture:

- **InceptionResNetV2 Encoder:** The InceptionResNetV2 model is a deep convolutional neural network (CNN) that has been pre-trained on the large-scale ImageNet dataset. It consists of a series of convolutional layers, pooling layers, and inception blocks. The InceptionResNetV2 architecture incorporates the concept of inception modules, which consist of parallel convolutional branches with different kernel sizes to capture both local and global features effectively. This encoder network serves as the feature extractor.
- **U-Net Decoder**: The decoder network extends from the last layer of the InceptionResNetV2 model and performs the upsampling and merging operations typical of the U-Net architecture. The decoder network gradually upsamples the feature maps to match the original input size while also fusing information from earlier layers to improve spatial resolution and localization accuracy.
- **Skip Connections:** One key aspect of the U-Net architecture is the inclusion of skip connections that connect the encoder and decoder networks at multiple spatial resolutions. These skip connections provide a shortcut for information flow, enabling the decoder to access feature maps from earlier layers of the encoder. This helps to preserve fine-grained details and facilitate more precise segmentation.

- **Concatenation:** In the InceptionResNetV2-UNet architecture, the feature maps from the encoder and decoder networks are concatenated at each corresponding spatial resolution. Concatenation combines the high-level features from the encoder with the upsampled features from the decoder, providing a rich representation that captures both local and global contextual information.

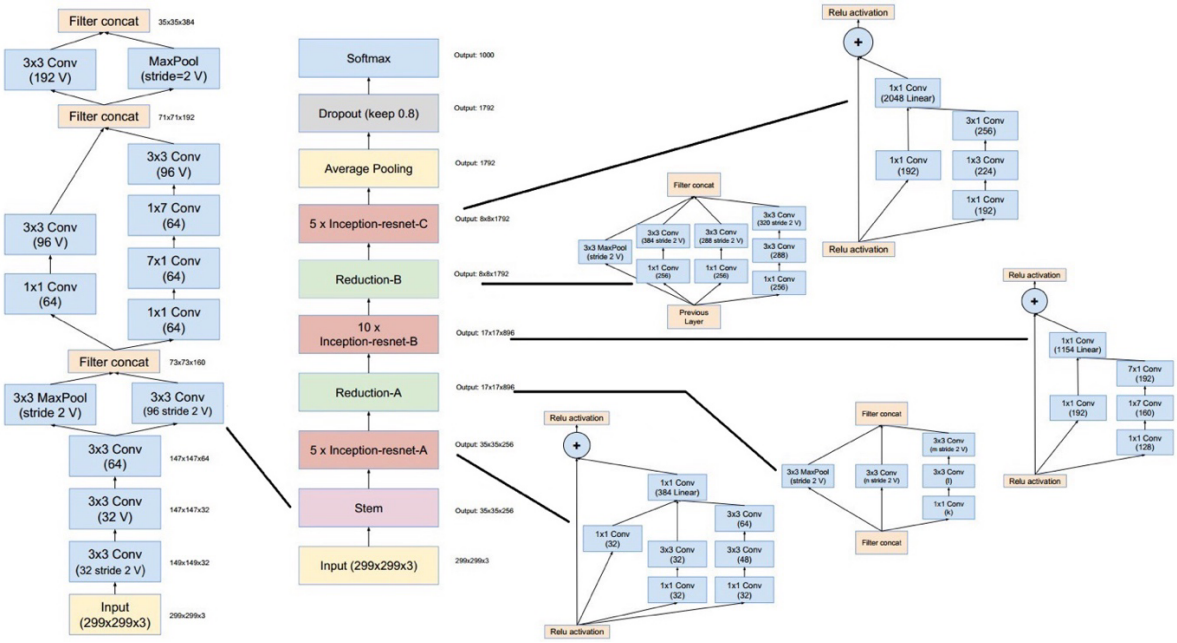


Figure 5 InceptionResNetV2-UNet Architecture

Methodology

This paper proposed improved semantic segmentation using InceptionResNetV2-UNet and CRF algorithms. The proposed method's workflow diagram is shown in Fig. 2. In addition, the proposed method includes a modified U-Net architecture with and without CRFs for "Aerial Image" to test both prediction accuracy and model complexity (in terms of time taken)

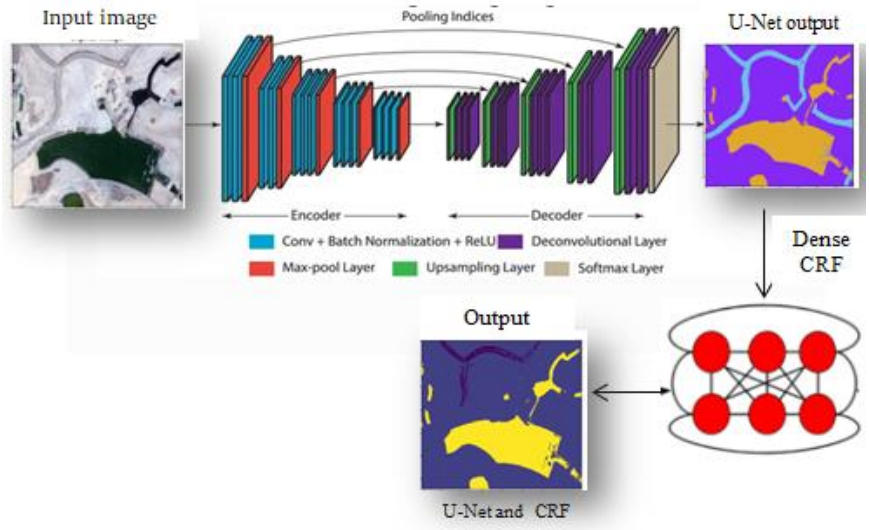
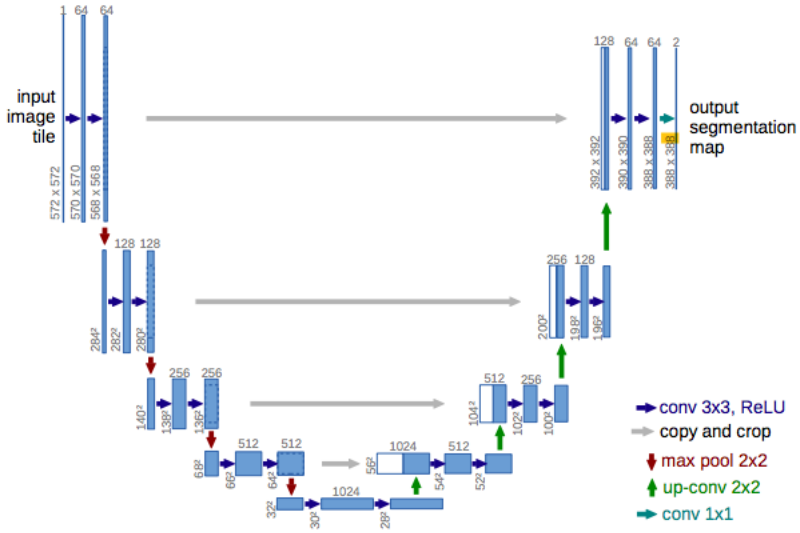


Fig.2: The Flowchart Diagram of Improved Proposed Method InceptionResNetV2-UNet

In the case of the InceptionResNetV2-UNet architecture, the goal is to connect the CRF layer to the original InceptionResNetV2-UNet model using PyTorch. PyTorch is a popular deep learning framework that provides tools for building and training neural networks. By leveraging PyTorch's capabilities, researchers can seamlessly integrate the CRF layer into the network architecture and train the entire model end-to-end.

The U-Net, originally developed by Olaf Ronneberger et al. [9], is a convolutional neural network primarily designed for biomedical image segmentation at the Computer Science Department of Freiburg. It has been further refined and optimized for accurate segmentation on modern GPUs, even with limited training data.

The U-Net architecture consists of encoder and decoder modules, each comprising two 3x3 convolutional layers. The encoder reduces the size of the feature map through max pooling after each convolutional layer. The decoder, on the other hand, employs bilinear interpolation to upsample the feature maps. Both encoder and decoder modules utilize batch normalization after each convolutional layer for improved performance, as illustrated in Figure 3.



As we progress from the initial encoder layers to the decoder, the precision of the feature maps decreases. To address this, it is necessary to incorporate features from previous encoder layers to enhance the information responsible for capturing fine details. This is accomplished using the InceptionResNetV2-UNet method, which establishes connections between encoder parts (rich in spatial information) and their corresponding decoders (rich in feature information).
The contracting path of the InceptionResNetV2-UNet consists of convolutional layers, with the number of channels increasing from one to 64 as the image depth grows during the convolution process. The red-colored arrows pointing downwards represent the max-pooling operation, which reduces the image size by half. The size drop from 572x572x568x568 is due to padding challenges, but "padding=same" is employed here.

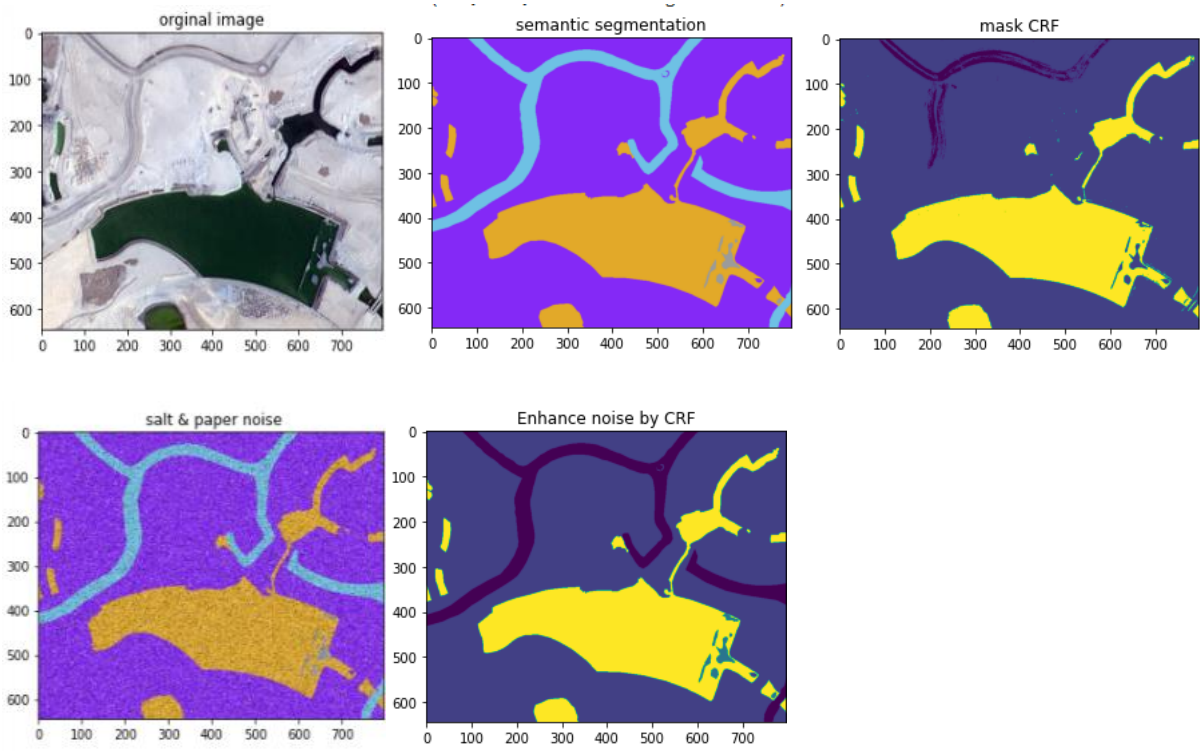
In the expanding path, the image is resized back to its original size using transposed convolution, a technique for upsampling images. The original image is padded before the convolution operation. After transposed convolution, the image size increases from 28x28x1024 to 56x56x512. This process combines information from previous layers to produce

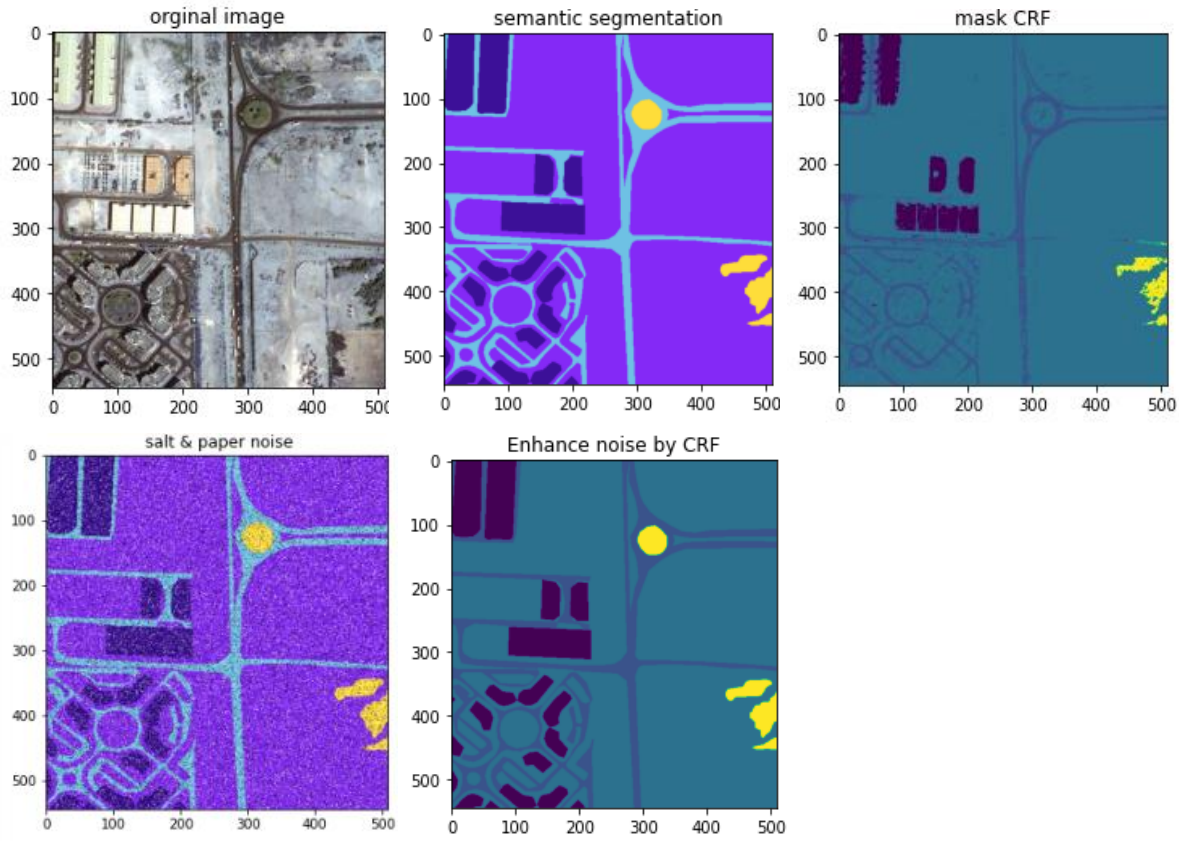
more accurate predictions and involves two additional convolution layers. This process is repeated three more times. Finally, the image is reshaped to meet the prediction requirements. The first layer employs a convolutional layer with 11 filters. Unlike conventional CNNs for classification tasks, a dense layer is not utilized throughout the network. The training process of the neural network remains the same.

The proposed method, known as InceptionResNetV2-UNet with CRF, leverages the encoder-decoder network, a popular deep learning architecture for semantic segmentation. The encoder utilizes convolutions, activation functions, and pooling operations to map the image space to a smaller latent space. The decoder then translates this latent space to the label space using transpose convolutions, activation functions, and upscaling layers. For aerial image segmentation, we adapt existing InceptionResNetV2-UNet architectures and further enhance segmentation accuracy by incorporating conditional random fields (CRFs).

Training and Evaluation

The InceptionResNetV2-UNet model was improved by the CRFs model for semantic segmentation of the aerial image database in Python. Good results were obtained, with the data training process taking 30 mints at epoch equal to 100, accuracy equal to 0.9639, and loss function equal to 0.5872. Compared to the not improved InceptionResNetV2-UNet i.e., without using CRFs, where the time to train the data was 4 hours, the accuracy value was 0.86. The loss function value was 0.66, and we achieved a great achievement by taking the least amount of time, having the highest accuracy, and having the least loss function.



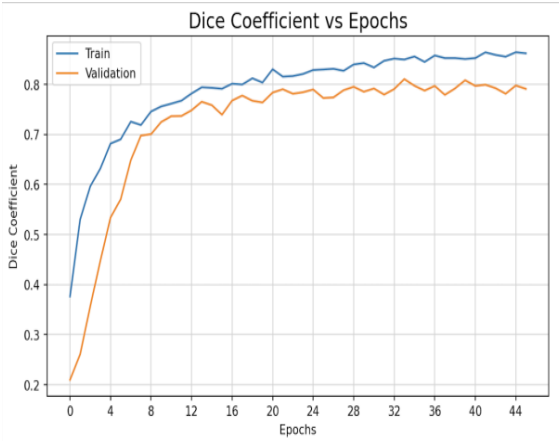


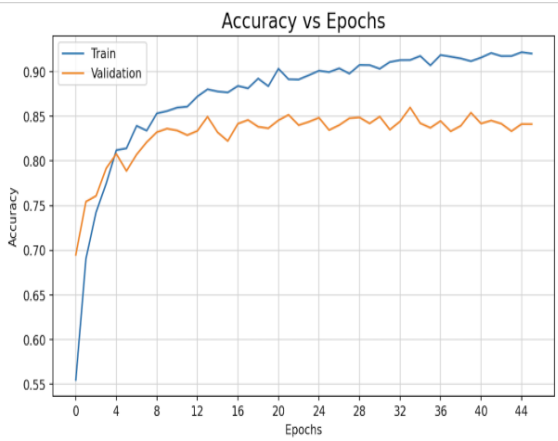
Hyper-Parameters

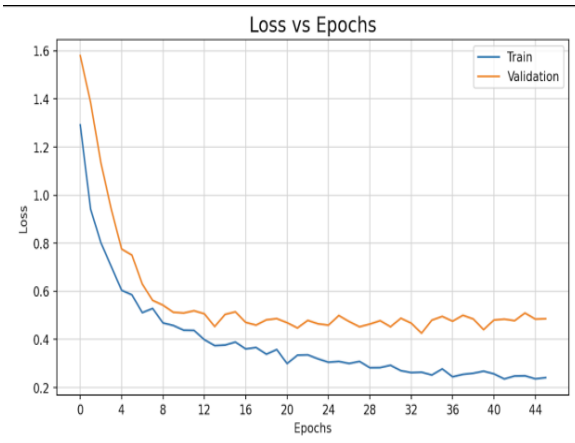
- Batch Size = 16.0
- Steps per Epoch = 32.0
- Validation Steps = 4.0
- Input Shape = (512, 512, 3)
- Initial Learning Rate = 0.0001 (with Exponential Decay LearningRateScheduler callback)
- Number of Epochs = 45 (with ModelCheckpoint & EarlyStopping callback)

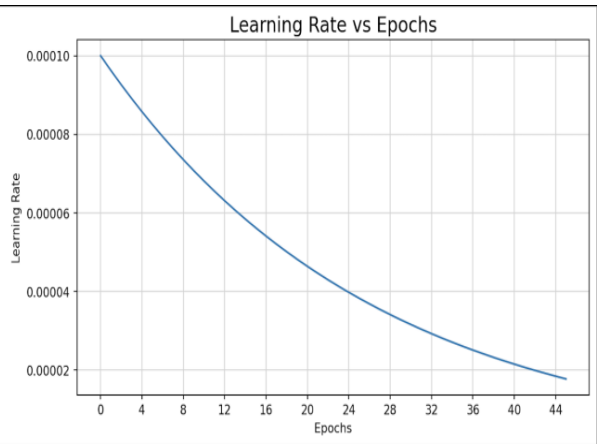
Training Results

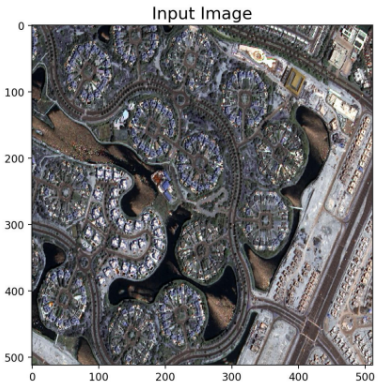
Model	Epochs	Train Dice Coefficient	Train Accuracy	Train Loss	Val Dice Coefficient	Val Accuracy	Val Loss
InceptionResNetV2-UNet	45 (best at 34thepoch)	0.8525	0.9152	0.2561	0.8112	0.8573	0.4268
InceptionResNetV2-Unet (With CRF)	100 (best at 93 epoch)	0.9025	0.9352	0.3561	0.852	0.9639	0.587

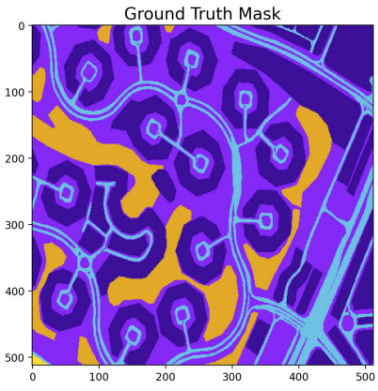

Dice Coefficient vs Epochs
Train
Validation
Dice Coefficient
Epochs

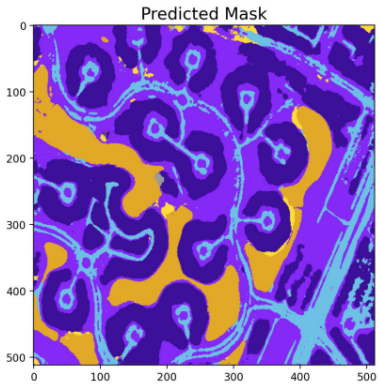

Accuracy vs Epochs
Train
Validation
Accuracy
Epochs

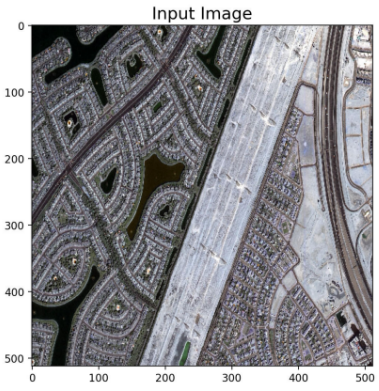

Loss vs Epochs
Train
Validation
Loss
Epochs

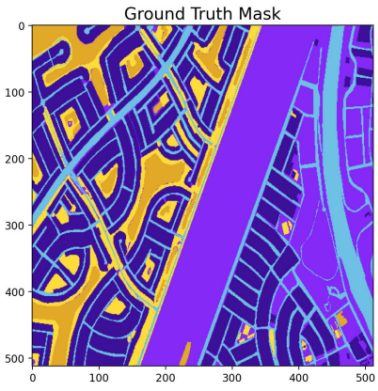

Learning Rate vs Epochs
Learning Rate
Epochs

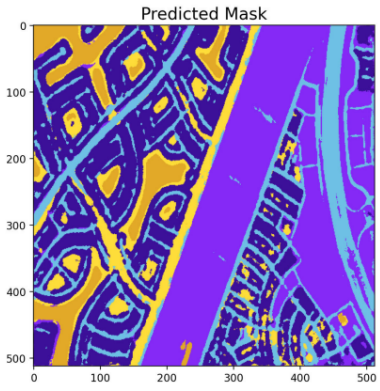

Input Image


Ground Truth Mask


Predicted Mask


Input Image


Ground Truth Mask


Predicted Mask

Advantages of InceptionResNetV2-UNet

- The InceptionResNetV2-UNet model combines the strengths of two popular architectures, InceptionResNetV2 and U-Net, and offers several advantages for image segmentation tasks:
- Feature Extraction: InceptionResNetV2 is a powerful deep convolutional neural network (CNN) known for its excellent feature extraction capabilities. It can capture intricate and hierarchical features from images, enabling the model to learn rich representations.
- Multi-Scale Context: The Inception module used in InceptionResNetV2 allows for the extraction of features at multiple scales. This is beneficial for semantic segmentation as it helps the model understand the context and capture both local and global information in the image.
- Skip Connections: The U-Net architecture is renowned for its skip connections that connect the encoder and decoder pathways. These connections enable the flow of low-level feature information from the encoder to the decoder, aiding in precise localization and preserving spatial details during upsampling.
- Semantic and Spatial Information: By combining the InceptionResNetV2 backbone with the U-Net structure, the model can leverage the semantic information learned by InceptionResNetV2 and the spatial details captured by U-Net. This combination enhances the model's ability to accurately segment objects in images.
- Efficient Training: InceptionResNetV2-UNet benefits from transfer learning, as InceptionResNetV2 is often pre-trained on large-scale datasets such as ImageNet. This pre-training allows the model to initialize with learned features and speeds up the training process, especially when the target dataset is limited.
- Performance: The combination of InceptionResNetV2 and U-Net has demonstrated strong performance in various image segmentation tasks. It has achieved state-of-the-art results in medical imaging, remote sensing, and other domains, showcasing its effectiveness in accurately segmenting objects of interest.
- Flexibility: The InceptionResNetV2-UNet model can be adapted and fine-tuned for different segmentation tasks and datasets. Its modular design allows for easy customization and incorporation of additional layers or techniques to improve performance.
- Overall, InceptionResNetV2-UNet offers the benefits of both InceptionResNetV2 and U-Net, providing a robust and effective architecture for semantic image segmentation. It combines powerful feature extraction, multi-scale context, skip connections, and efficient training, resulting in accurate and detailed segmentation results.

Conclusion

The InceptionResNetV2-UNet with CRF approach utilizes the U-Net architecture, which has been successfully applied to various image segmentation tasks. It enables assigning different colors to each class and accurately categorizing pixels in the image, such as labeling them as cars or planes, achieving semantic segmentation. Unlike, FCN, the InceptionResNetV2-UNet with CRF approach incorporates copying and clipping operations, preserving spatial information and enhancing the model's performance. It outperforms previous methods, including the Fully Convolutional Network (FCN), in terms of accuracy and efficiency.

One of the advantages of the InceptionResNetV2-UNet with CRF approach is its speed. The segmentation of a 644 x 797 x 3 image can be done in less than a second on a modern GPU. This makes it suitable for real-time or near real-time applications.
Additionally, the proposed algorithm improves the U-Net architecture by incorporating a Conditional Random Field (CRF) post-processing or end-to-end refinement. The CRF helps in smoothing and refining pixel predictions in aerial image semantic segmentation. By combining the strengths of InceptionResNetV2-UNet and CRF, the algorithm achieves higher quality results, even in the presence of salt and pepper noise.

The comprehensive training system integrates the InceptionResNetV2-UNet with CRF into one deep network. This approach demonstrates superior performance compared to the original U-Net without CRF. It achieves excellent results with the least amount of time, highest accuracy, and lowest loss function, showcasing the effectiveness of the InceptionResNetV2-UNet with CRF approach in aerial image segmentation.