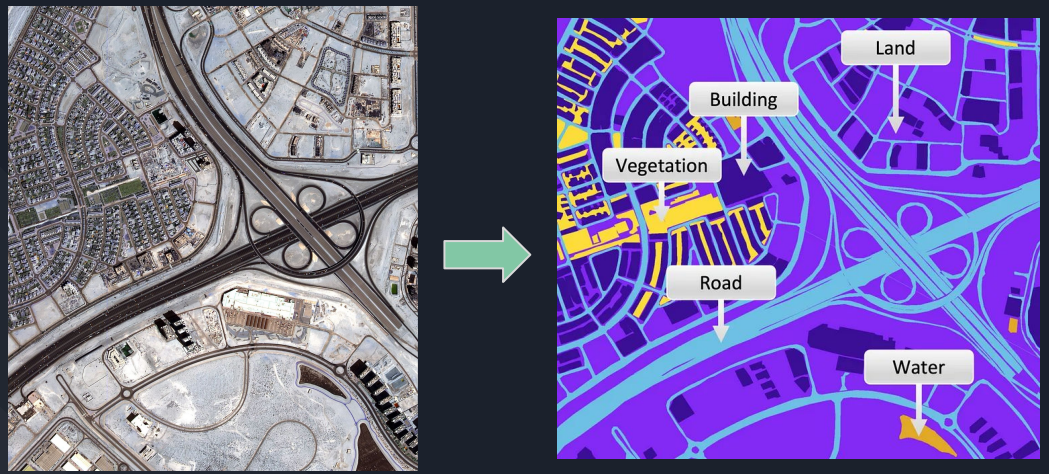
Semantic Segmentation on Aerial(Satellite) Images using U-Net with Hybrid CNN Models



Source:https://towardsdatascience.com/semantic-segmentation-of-aerial-imagery-using-u-net-in-python-552705238514

Presented by:
Sanketh Karuturi

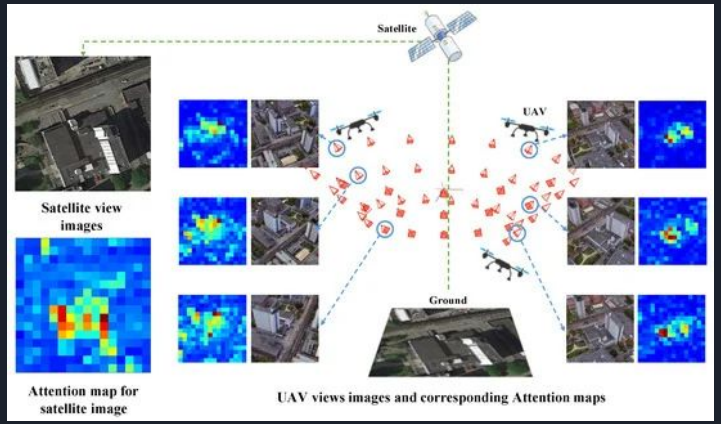
Problem Statement

The goal of this project is to perform semantic segmentation on aerial (satellite) images using the U-Net architecture with various Hybrid CNN Models. The challenge lies in accurately segmenting various land cover types, such as buildings, roads, vegetation, and water bodies, from high-resolution satellite imagery. By leveraging U-Net's encoder-decoder structure, the model aims to produce pixel-wise classifications, enabling precise delineation of objects for applications in urban planning, environmental monitoring, and disaster management.

Applications

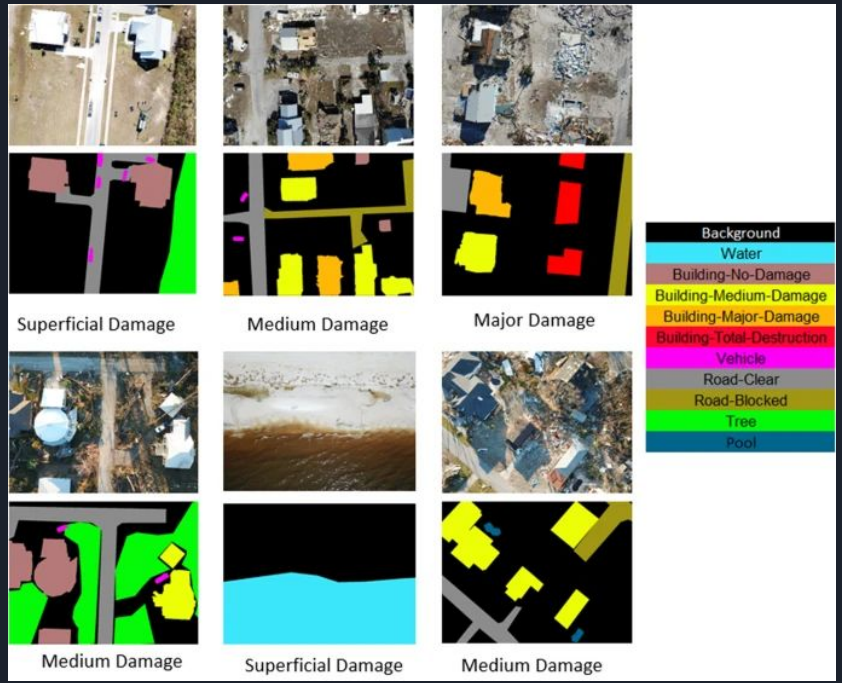
Urban Planning

Autonomous Navigation : UAV (drone) path



(Bouguettaya, A., 2022)

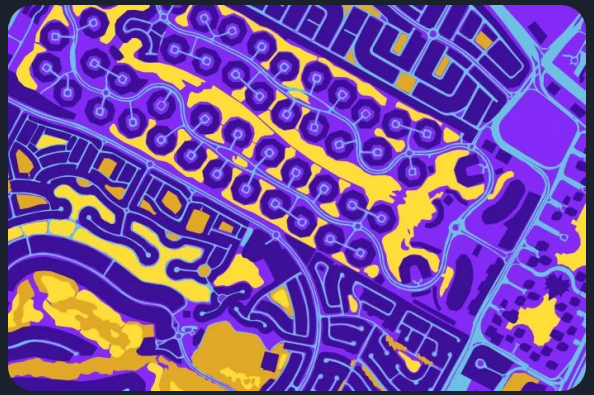
Disaster Management



(Rahnemoonfar, M, 2023)

Datasets

- Dubai MBRSC Dataset:
- This dataset consists of high-resolution satellite images of Dubai, UAE, provided by the Mohammed Bin Rashid Space Center (MBRSC). It has been pixel-wise annotated for semantic segmentation and is publicly available for research purposes.
- Dataset Composition
  - The dataset contains 72 satellite images, organized into six larger tiles.
  - Each image is annotated at the pixel level into six distinct classes:
    - Buildings (#3C1098)
    - Unpaved Land (#8429F6)
    - Roads (#6EC1E4)
    - Vegetation (#FEDD3A)
    - Water Bodies (#E2A929)
    - Unlabeled Areas (#9B9B9B)



Dubai MBRSC Dataset (Humans in the Loop, n.d.)

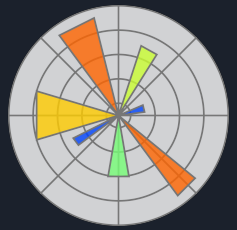
Technical Approach

1. Data Preprocessing – Image resizing, normalization, and augmentation.
2. Model Selection – Implement U-Net with Hybrid CNN Models.
3. Training – Optimize hyperparameters, loss functions, and learning rates.
4. Evaluation – Metrics like Validation Accuracy and IoU (Intersection over Union).
5. Deployment – Predict segmentation maps on new satellite images.

Technologies used:

Keras

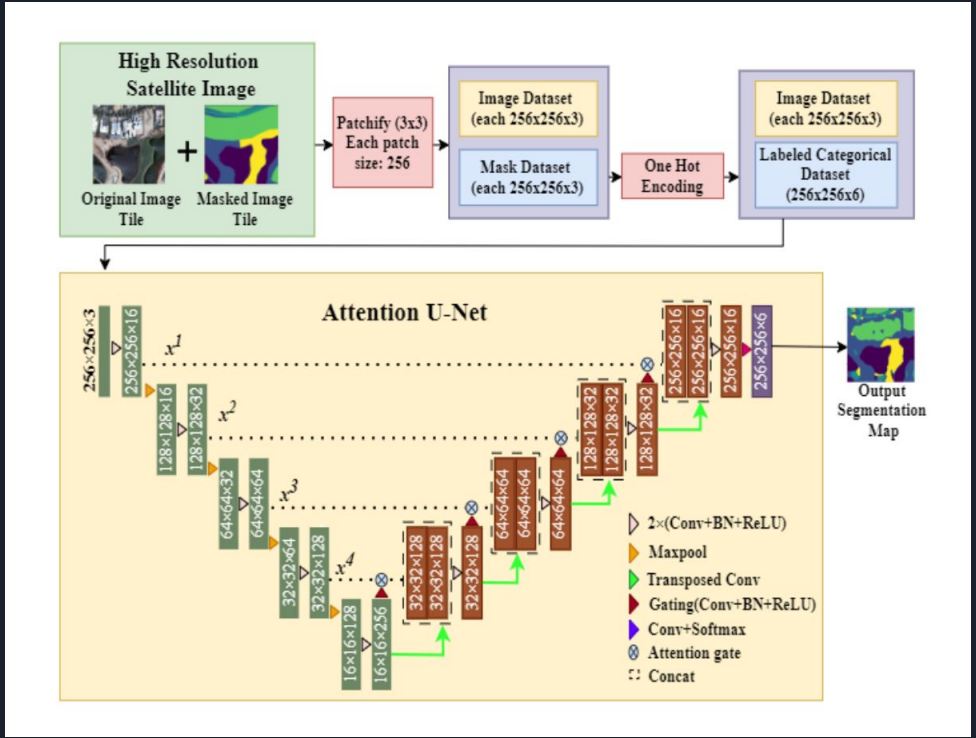
NumPy



OpenCV

Data Preprocessing & Augmentation

1) **Patch Extraction**
   - Extract patches from images and masks using the patchify library.
2) **Patch Resizing**
   - Resize all patches to 256×256 pixels for consistency.
3) **Min-Max Scaling**
   - Normalize pixel values to a range of 0 to 1.
4) **Mask Color Conversion**
   - Convert hexadecimal color values in masks to RGB format.
5) **Dataset Splitting**
   - 90% for training (X_train and Y_train)
   - 10% for testing X_test and Y_test.



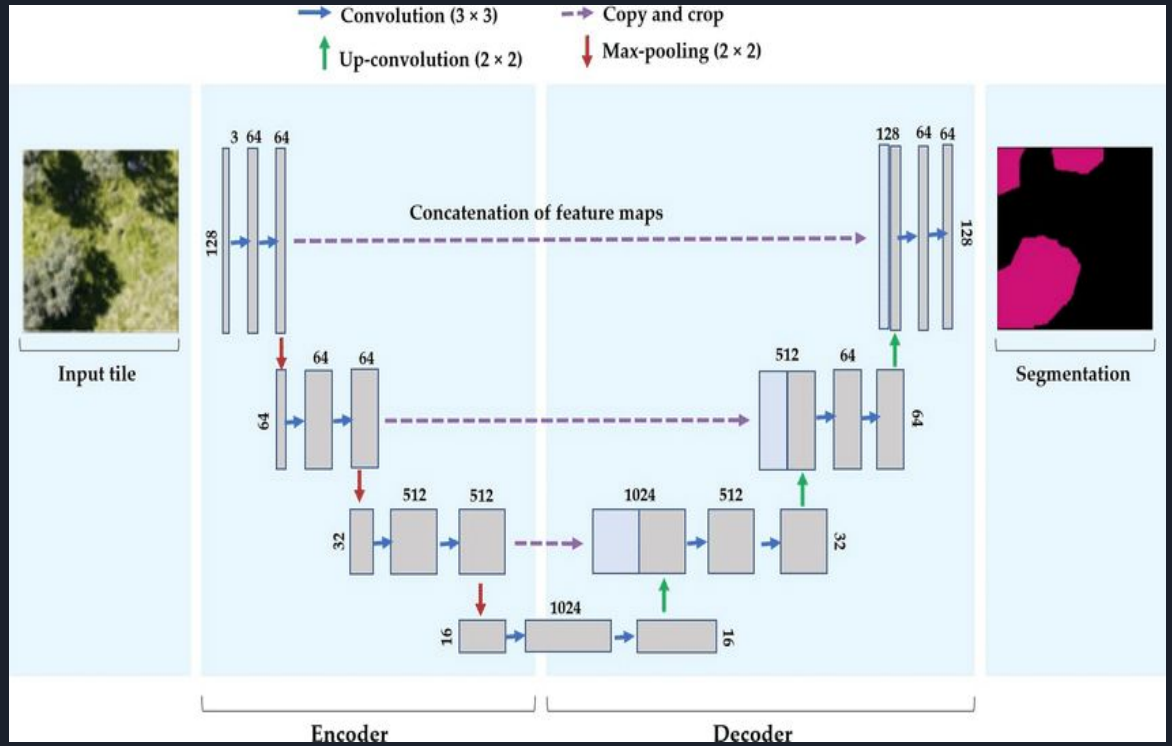
Example of semantic segmentation of an RGB satellite image (Islam et al., 2024).

U-Net Architecture

- Fully Convolutional Network (FCN) designed for precise segmentation.
- Encoder-Decoder structure for extracting global and local features.
- Skip connections help retain spatial information.

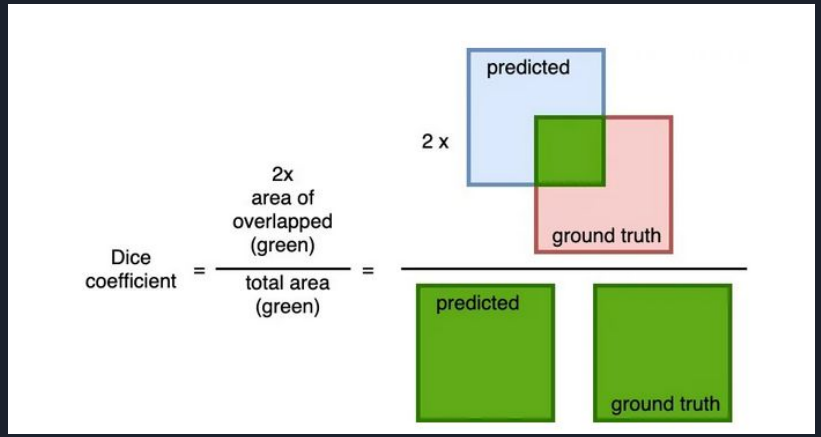
Architecture Breakdown:

- Encoder: Downsamples input image using CNN layers.
- Bottleneck: Extracts deep feature representations.
- Decoder: Upsamples to restore image resolution.
- Skip Connections: Merge encoder features to refine segmentation.



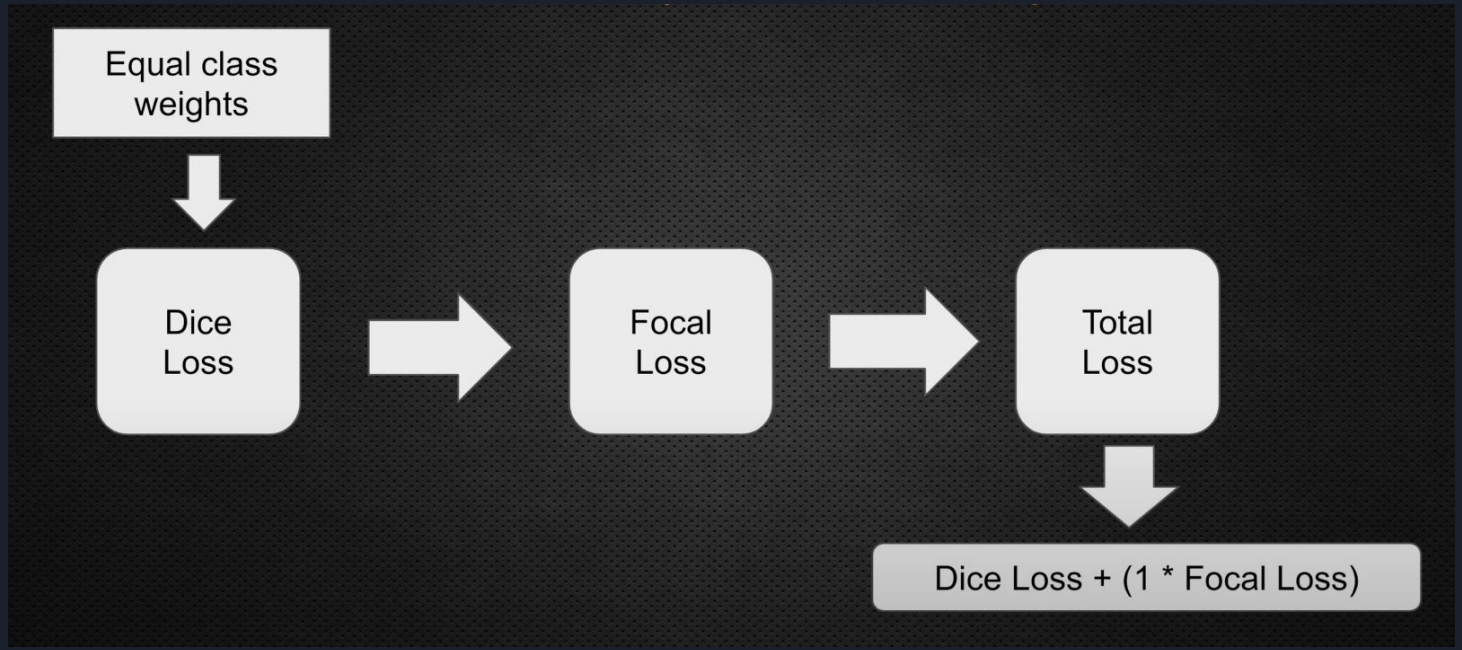
Simplified architecture of the U-net Model for Segmentation of Satellite Images (Bouguettaya et al., 2022)

Dice Loss



$$DL = 1 - \frac{1}{2} \sum_{c=1}^{2} \frac{2 \sum_{i=1}^{N} x_{c,i} y_{c,i}}{\sum_{i=1}^{N} x_{c,i}^2 + \sum_{i=1}^{N} y_{c,i}}$$

Loss Function : Focal Loss(Cross Entropy Loss Extension)



$$FL = -(1 - P_t)^{\gamma} \log(P_t)$$

$$CE = -\log(P_t)$$

$$FocalLoss = -\sum_{i=1}^{i=n} (i - p_i)^{\gamma} log_b(p_i)$$

Training & Evaluation

1) In image segmentation, a standard success indicator is the Intersection Over Union (IOU) coefficient also called Jaccard Index.

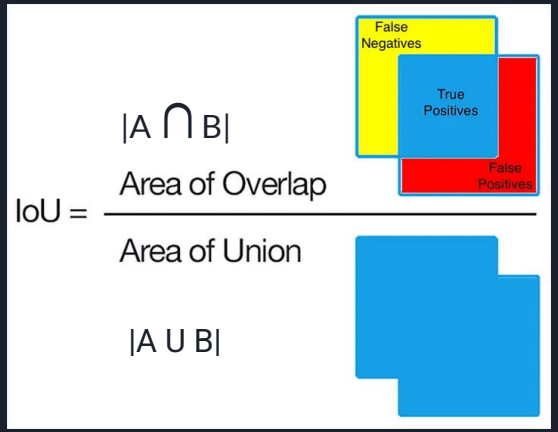


Illustration of the Jaccard index calculation (Wikipedia contributors, 2025).

2) IOU or Jaccard Index measures the number of overlapping pixels enclosed by the actual and predicted masks divided by the total pixels across both masks

Callbacks

Specific callbacks will be helpful when we train a neural network as they offer a *view of the model's internal state during training*. Three callbacks are:

1. ModelCheckpoint: saves the best model out of all epochs with the highest validation accuracy

2. EarlyStopping: stops fitting if validation_loss does not continue to decrease over two epochs

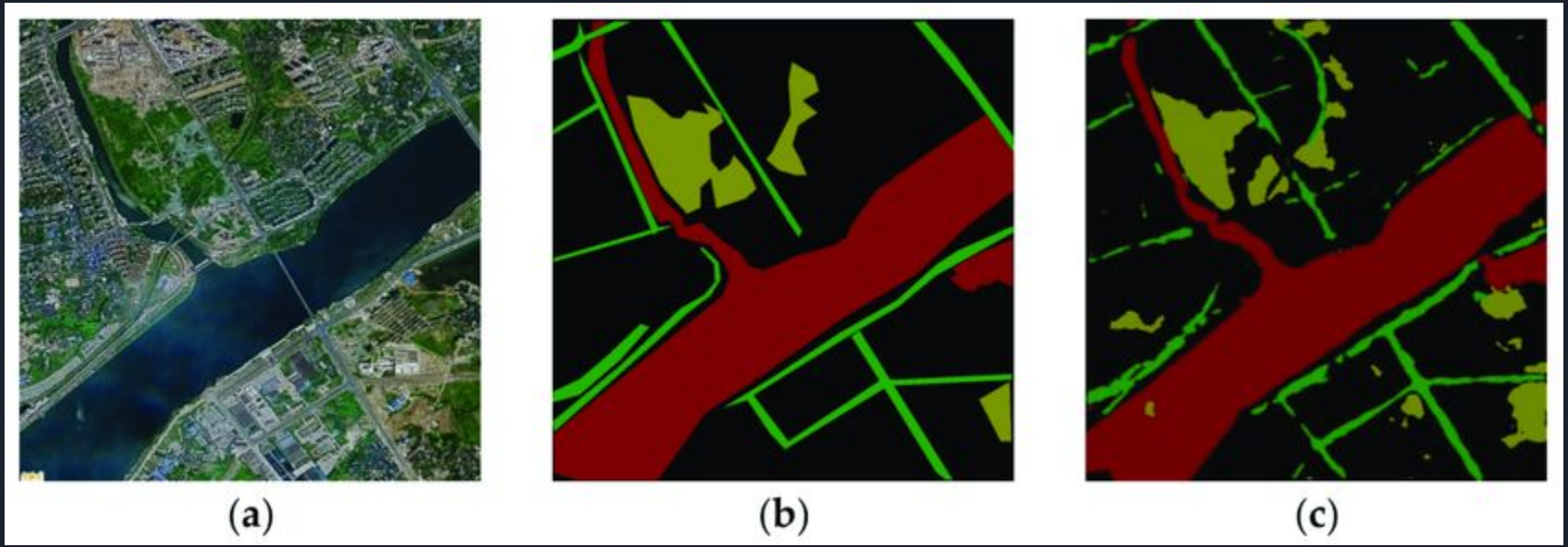
3. CSVLogger: saves model state values to a CSV file on each epoch

Once the callbacks are defined, the model compiles with Adam *optimisation*, categorical_crossentropy *loss* and training initiates for *X epochs and Batch_Size = Y*

After training completion, the model is saved to the hard disk

Results : Prediction on Custom Images

INPUT GROUND TRUTH PREDICTION



(Tong et al., 2023).

1) Standard Vanilla U-Net

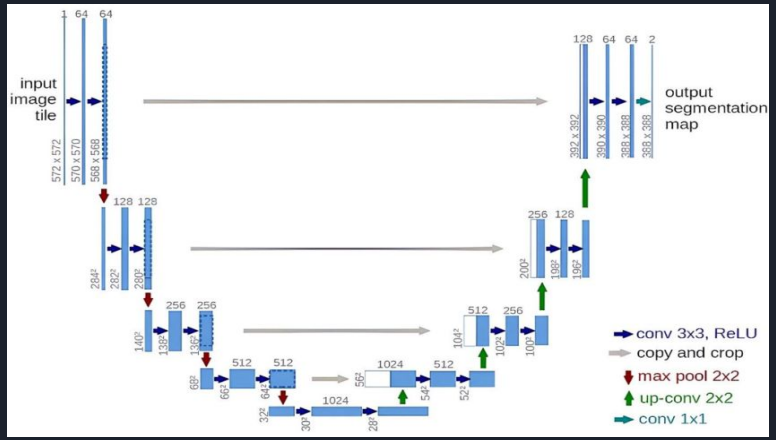


Fig 1.1 Standard Vanilla U-Net Architecture (Rahman & Mahanta, 2024)

Model	Standard Vanilla U-Net
Model Size (MB)	7.41
Total Number of Parameters	1,941,190
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	85.90%
Peak Validation_Jaccard_Coeff	0.7428
mIOU	0.5966

Table 1.1 Standard Vanilla U-Net Performance Metrics

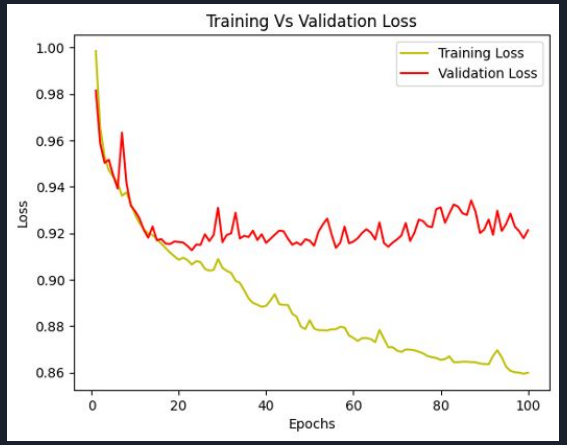


Fig 1.2. Standard Vanilla U-Net Training Vs Validation Loss Graph

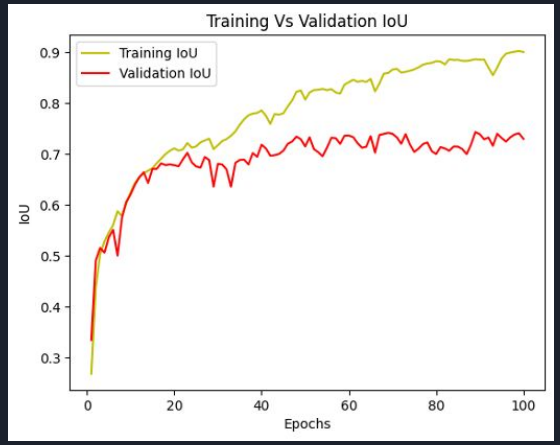


Fig 1.3. Standard Vanilla U-Net Training Vs Validation IoU Graph

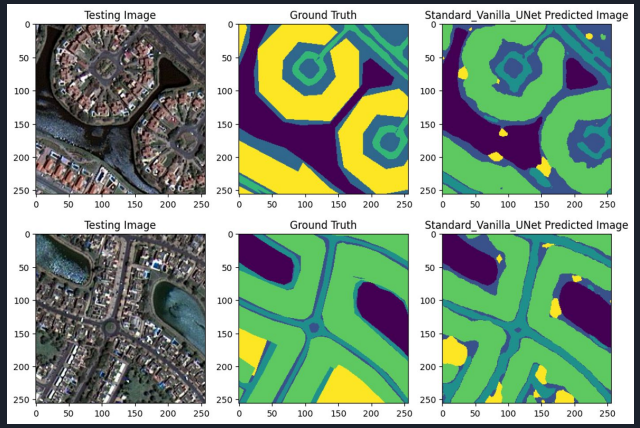


Fig 1.4. Standard Vanilla U-Net Model Predicted Images

1.1) Standard Vanilla U-Net with Dropout (0.1)

Model	Vanilla U-Net, Dropout=0.1
Model Size (MB)	7.41
Total Number of Parameters	1,941,190
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	87.31%
Peak Validation_Jaccard_Coeff	0.7605
mIOU	0.5982

Table 1.1.1. Standard Vanilla U-Net with Dropout (0.1) Performance Metrics

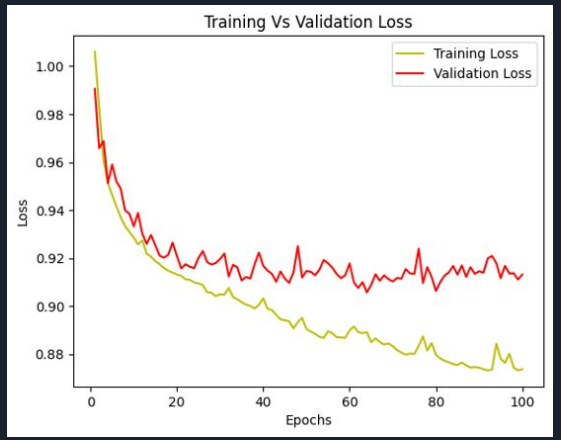


Fig 1.1.1. Standard Vanilla U-Net with Dropout(0.1) Training Vs Validation Loss Graph

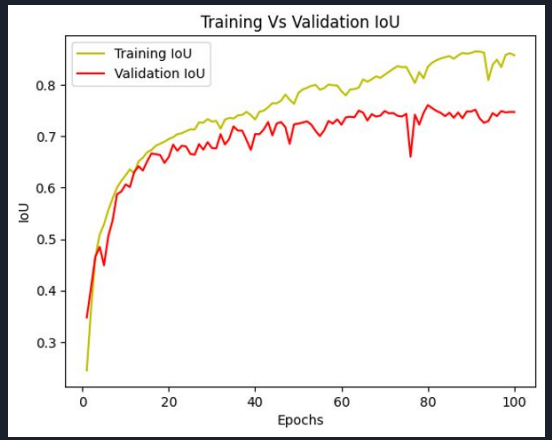


Fig 1.1.2. Standard Vanilla U-Net with Dropout(0.1) Training Vs Validation IoU Graph

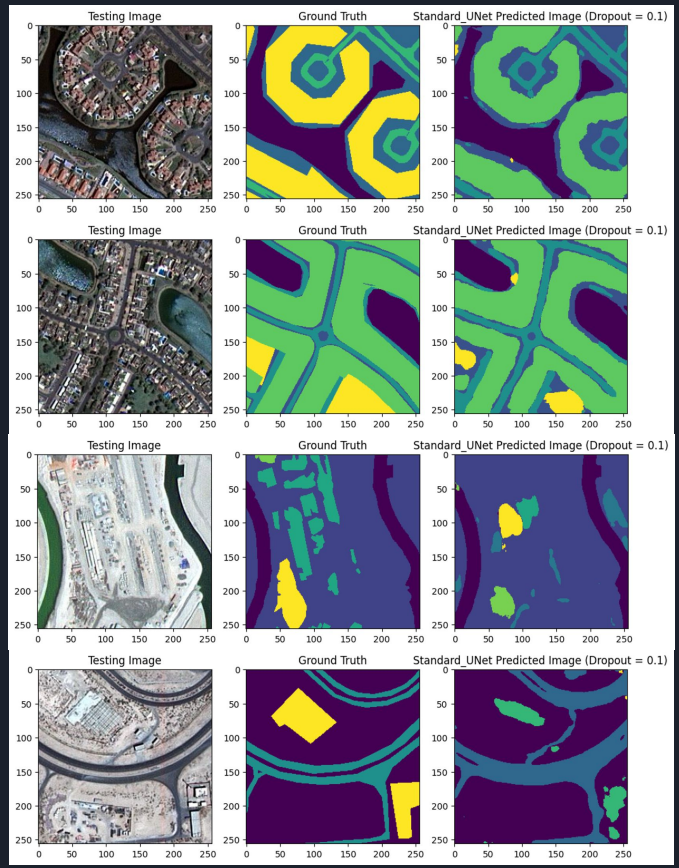


Fig 1.1.3. Standard Vanilla U-Net with Dropout(0.1) Model Predicted Images

1.2) Standard Vanilla U-Net with Dropout (0.2)

Model	Vanilla U-Net, Dropout=0.2
Model Size (MB)	7.41
Total Number of Parameters	1,941,190
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	86.59%
Peak Validation_Jaccard_Coeff	0.7458
mIOU	0.5942

Table 1.2.1. Standard Vanilla U-Net with Dropout (0.2) Performance Metrics

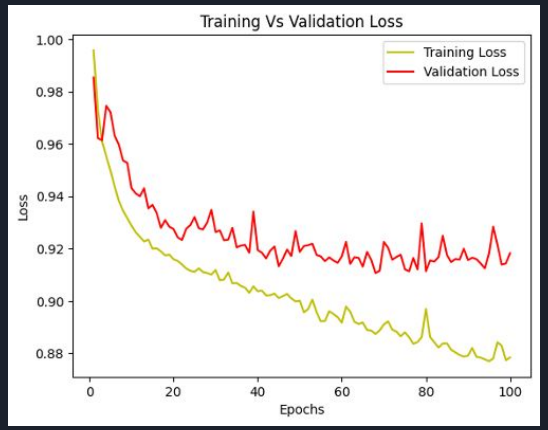


Fig 1.2.1. Standard Vanilla U-Net with Dropout(0.2) Training Vs Validation Loss Graph

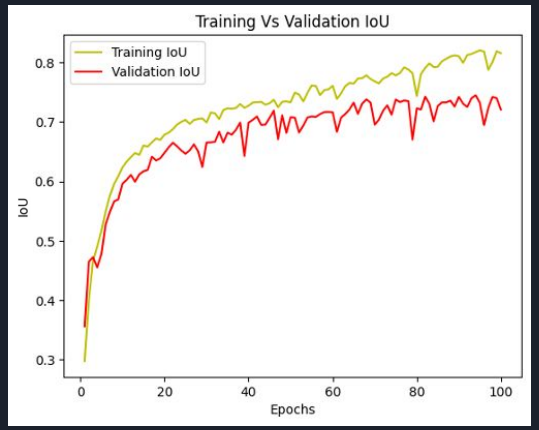


Fig 1.2.2. Standard Vanilla U-Net with Dropout(0.2) Training Vs Validation IoU Graph

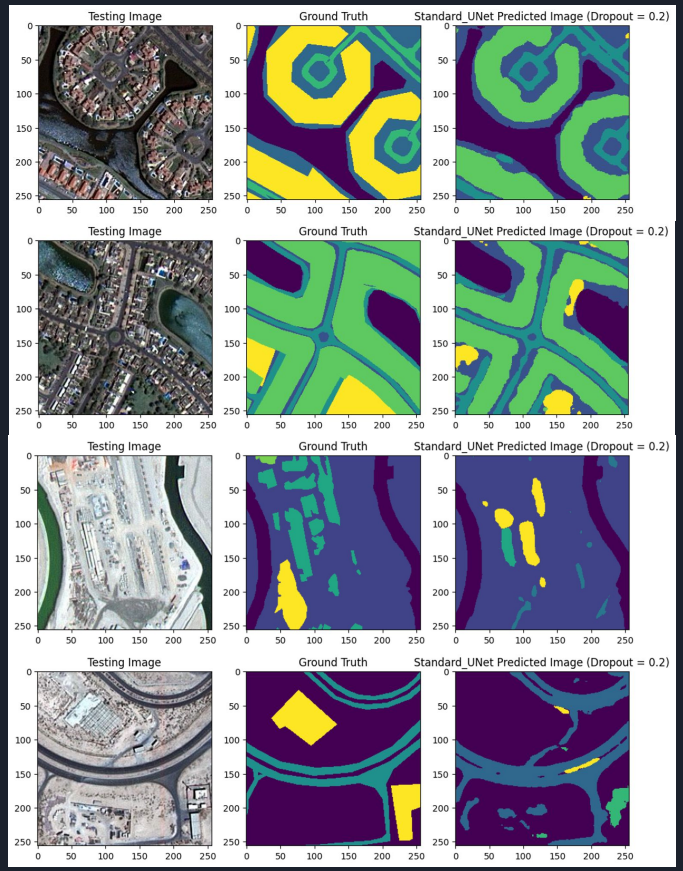


Fig 1.2.3. Standard Vanilla U-Net with Dropout(0.2) Model Predicted Images

1.3) Standard Vanilla U-Net with Dropout (0.3)

Model	Vanilla U-Net, Dropout=0.3
Model Size (MB)	7.41
Total Number of Parameters	1,941,190
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	85.90%
Peak Validation_Jaccard_Coeff	0.7334
mIOU	0.5843

Table 1.3.1. Standard Vanilla U-Net with Dropout (0.3) Performance Metrics

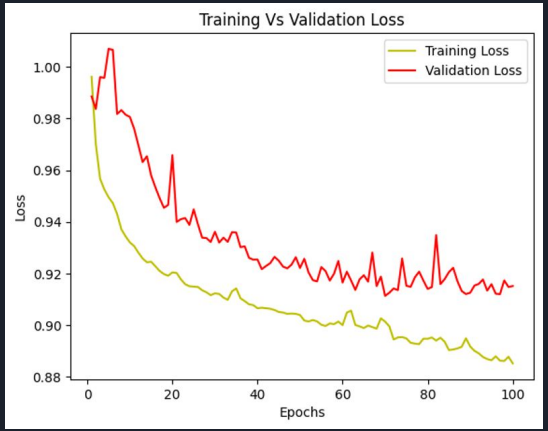


Fig 1.3.1. Standard Vanilla U-Net with Dropout(0.3) Training Vs Validation Loss Graph

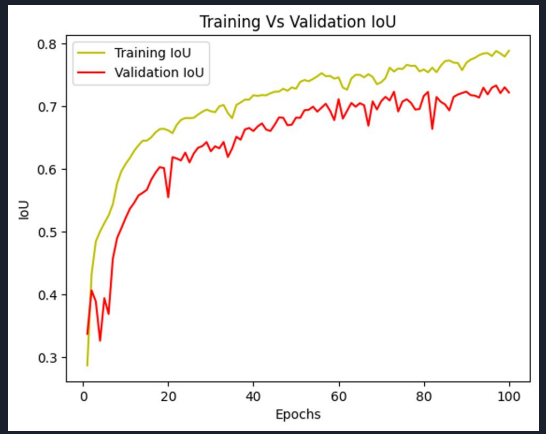


Fig 1.3.2. Standard Vanilla U-Net with Dropout(0.3) Training Vs Validation IoU Graph

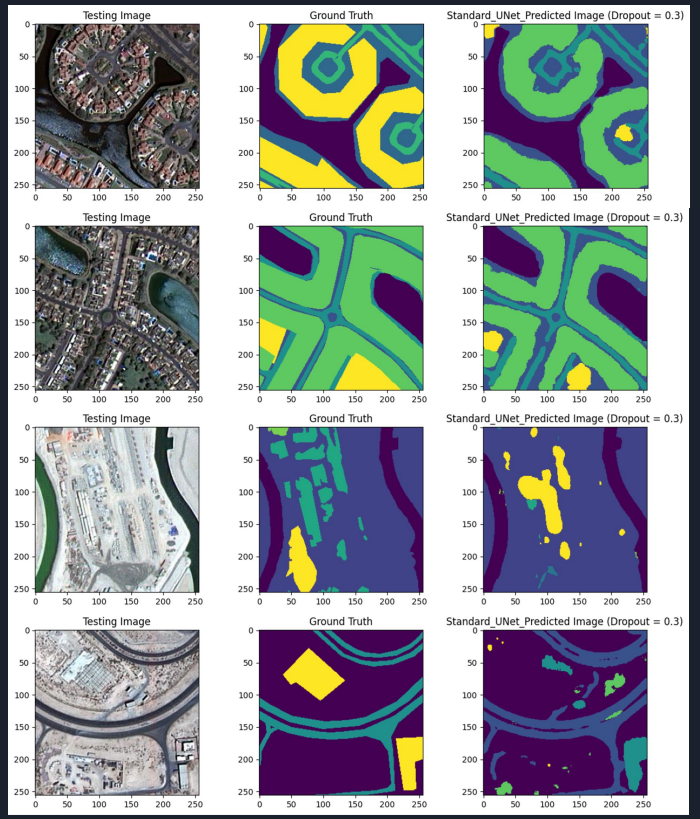


Fig 1.3.3. Standard Vanilla U-Net with Dropout(0.3) Model Predicted Images

Dropout Predictions Visual Comparison

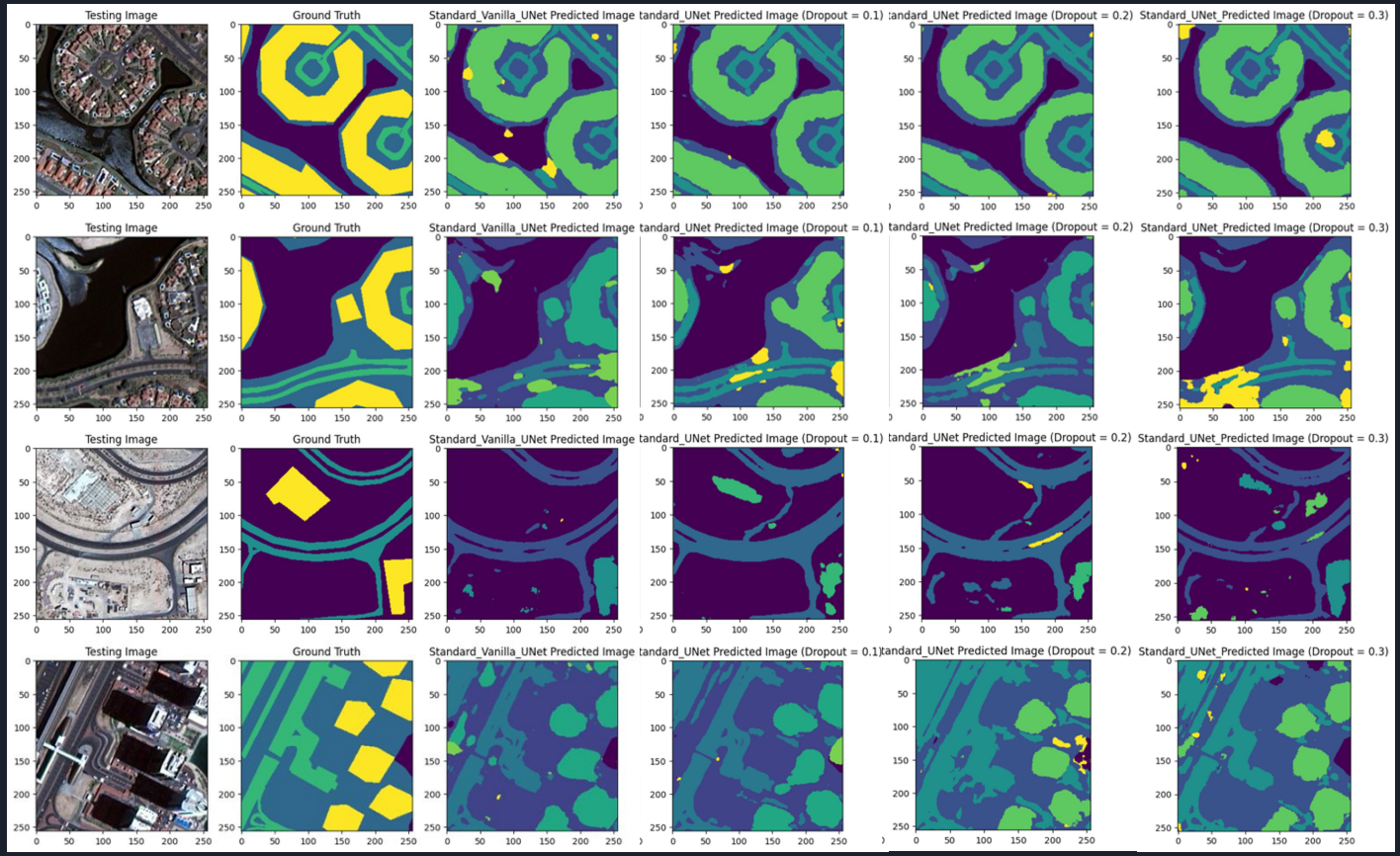


Fig 1.2. Standard Vanilla U-Net Dropout Predictions Visual Comparison, Dropouts = 0.1, 0.2, 0.3

2) VGG16 U-Net

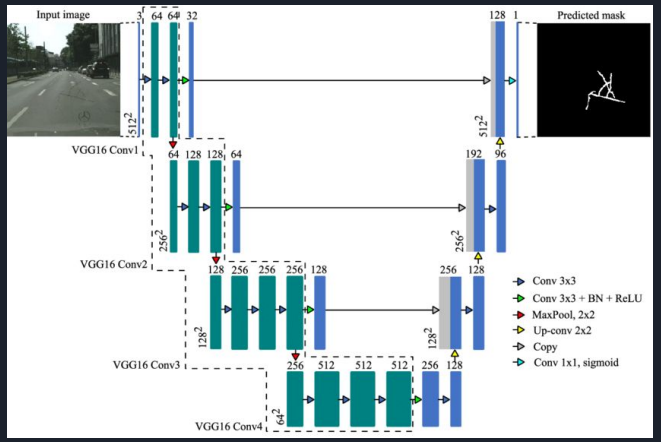


Fig 2.1. VGG16 + U-Net Architecture (Kanaeva & Ivanova, 2021)

Model	VGG16 U-Net
Model Size (MB)	98.66
Total Number of Parameters	25,862,662
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	88.10%
Peak Validation_Jaccard_Coeff	0.7697
mIOU	0.6059

Table 2.1. VGG16 + U-Net Performance Metrics

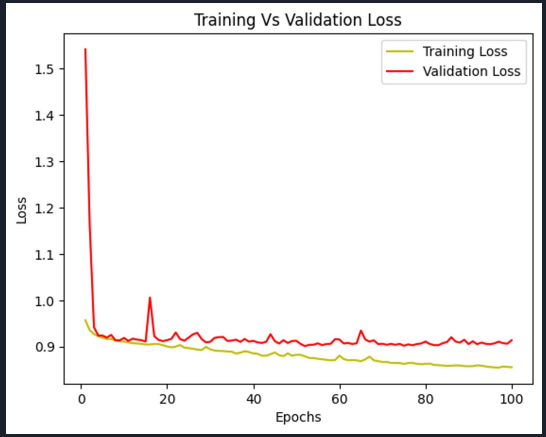


Fig 2.2. VGG16 + U-Net Training Vs Validation Loss Graph

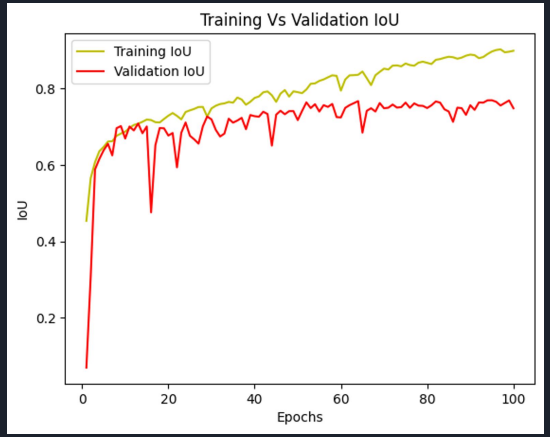


Fig 2.3. VGG16 + U-Net Training Vs Validation IoU Graph

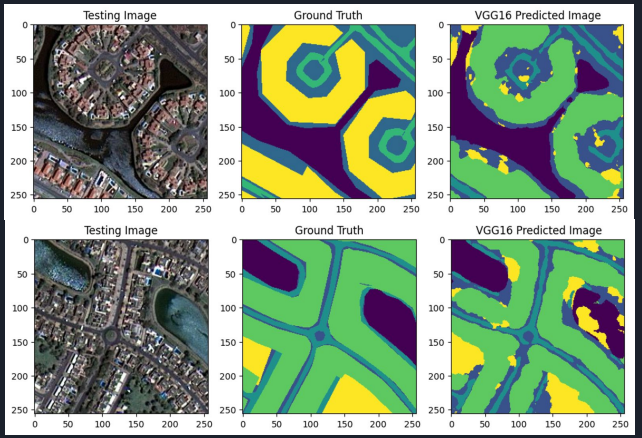


Fig 2.4. VGG16 + U-Net Model Predicted Images

3) VGG19 U-Net

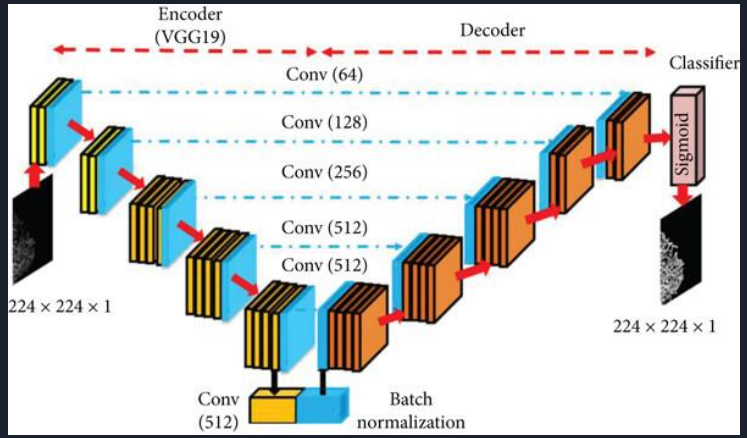


Fig 3.1. VGG19 + U-Net Architecture (Daniel et al., 2022)

Model	VGG19 U-Net
Model Size (MB)	118.91
Total Number of Parameters	31,172,358
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	88.03%
Peak Validation_Jaccard_Coeff	0.7443
mIOU	0.5524

Table 3.1. VGG19 + U-Net Performance Metrics

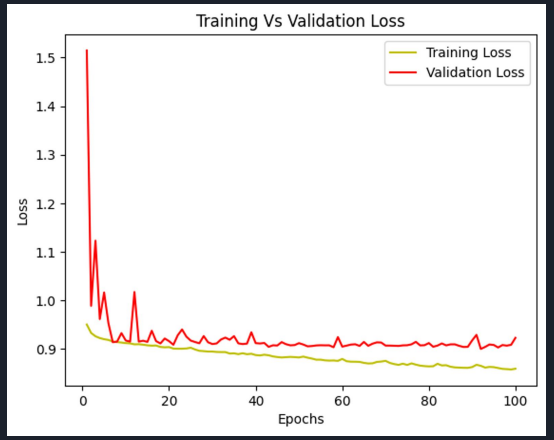


Fig 3.2. VGG19 + U-Net Training Vs Validation Loss Graph

Fig 3.3. VGG19 + U-Net Training Vs Validation IoU Graph

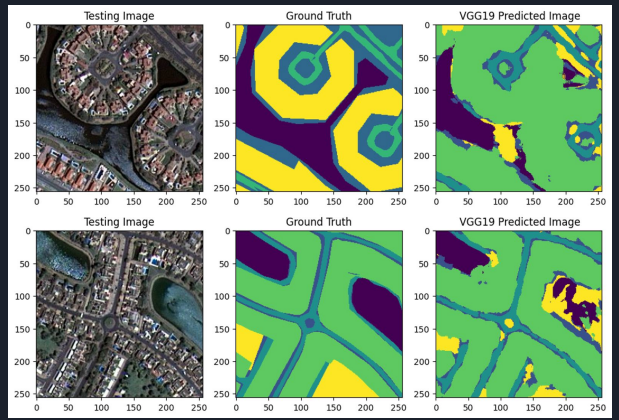


Fig 3.4. VGG19 + U-Net Model Predicted Images

4) ResNet18 U-Net

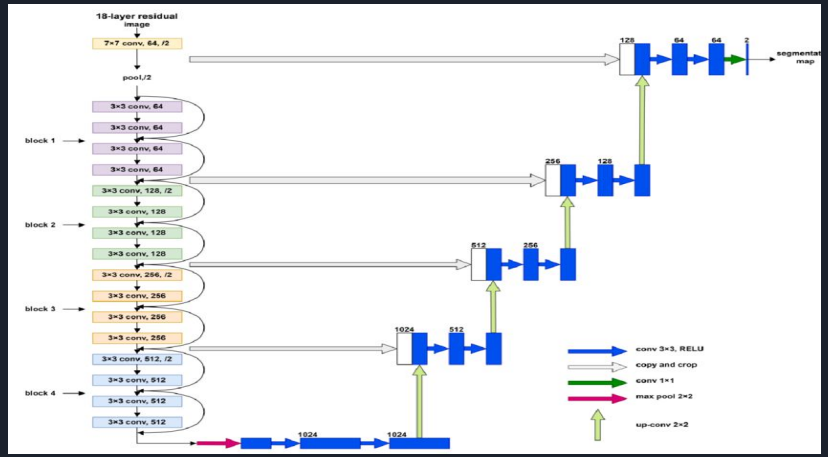


Fig 4.1. ResNet18 U-Net Architecture (Chen, 2023)

Model	ResNet18 U-Net
Model Size (MB)	54.71
Total Number of Parameters	14,341,295
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	87.54%,
Peak Validation_Jaccard_Coeff	0.7622
mIOU	0.603

Table 4.1. ResNet18 U-Net Performance Metrics

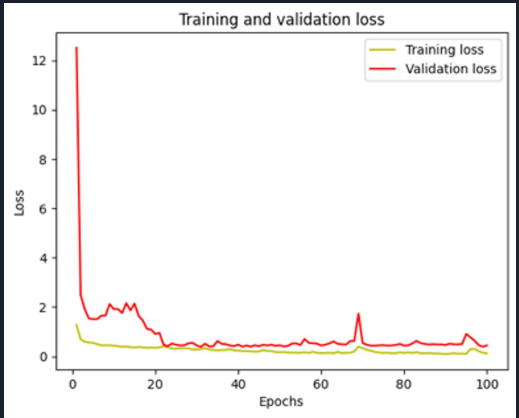


Fig 4.2. ResNet18 U-Net Training Vs Validation Loss Graph

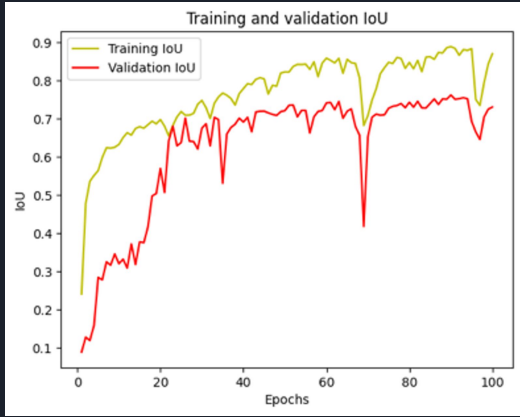


Fig 4.3. ResNet18 U-Net Training Vs Validation IoU Graph

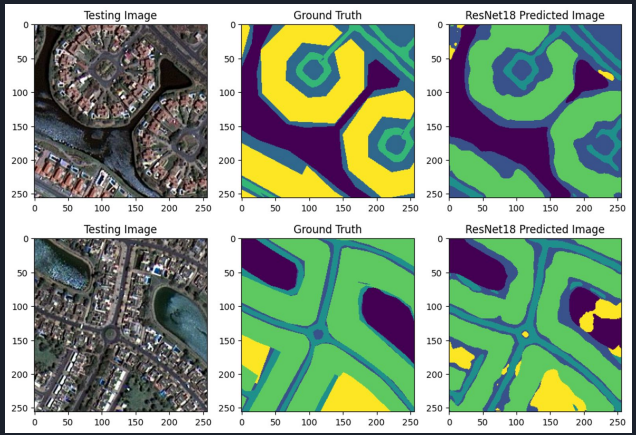


Fig 4.4. ResNet18 U-Net Model Predicted Images

5) ResNet34 U-Net

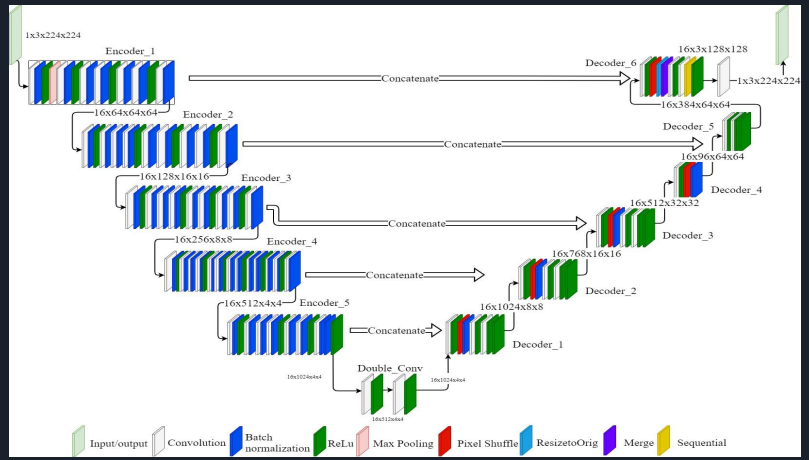


Fig 5.1. ResNet34 U-Net Architecture

Model	ResNet34 U-Net
Model Size (MB)	93.30
Total Number of Parameters	24,456,879
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	87.88%
Peak Validation_Jaccard_Coeff	0.7622
mIOU	0.63

Table 5.1. ResNet34 U-Net Performance Metrics

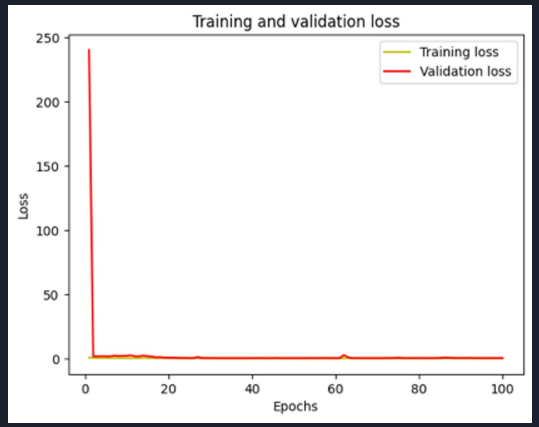


Fig 5.2. ResNet34 U-Net Training Vs Validation Loss Graph

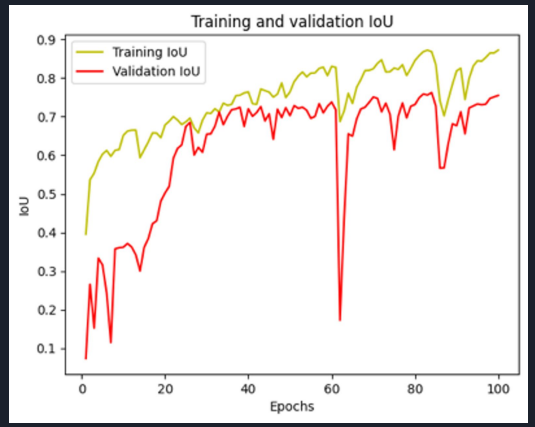


Fig 5.3. ResNet34 U-Net Training Vs Validation IoU Graph

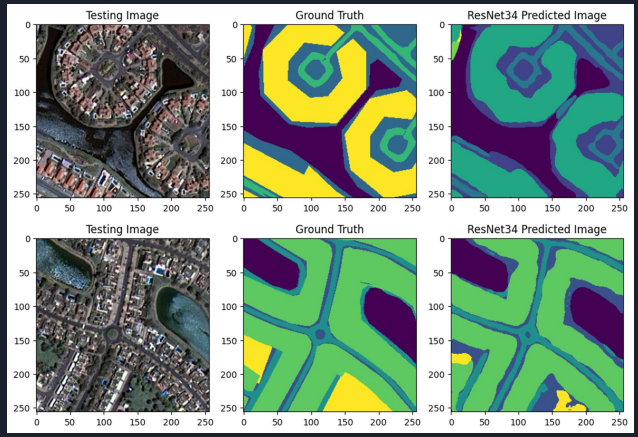


Fig 5.4. ResNet34 U-Net Model Predicted Images

6) ResNet50 U-Net

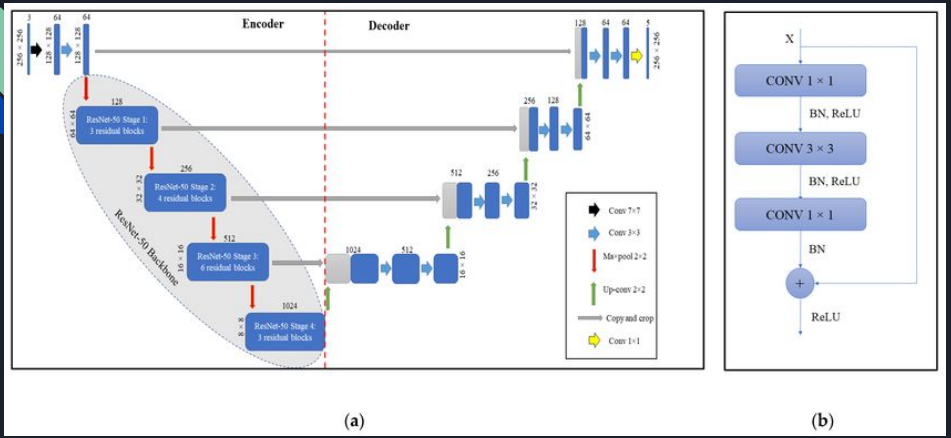


Fig 6.1. ResNet50 U-Net Architecture (Manos et al., 2022)

Model	ResNet50 U-Net
Model Size (MB)	124.21
Total Number of Parameters	32,561,839
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	87.98%,
Peak Validation_Jaccard_Coeff	0.7590
mIOU	0.619

Table 6.1. ResNet50 U-Net Performance Metrics

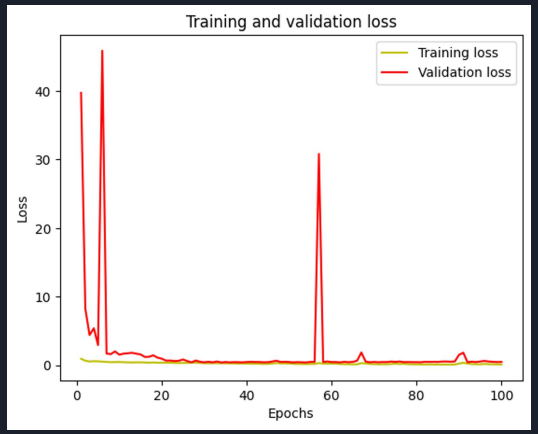


Fig 6.2. ResNet50 U-Net Training Vs Validation Loss Graph

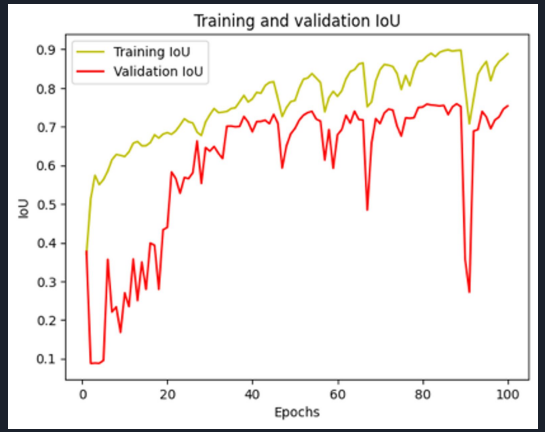


Fig 6.3. ResNet50 U-Net Training Vs Validation IoU Graph

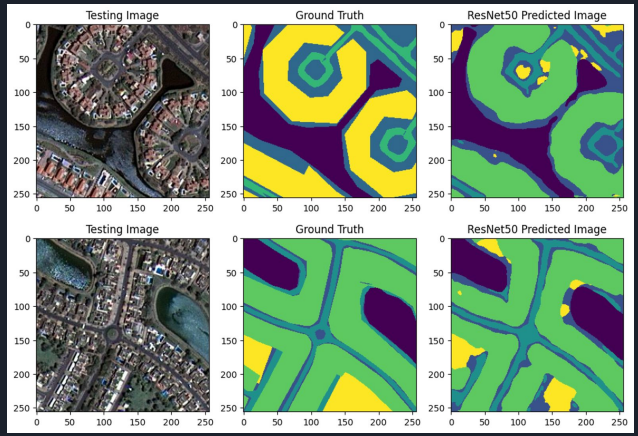


Fig 6.4. ResNet50 U-Net Model Predicted Images

7) ResNet101 U-Net

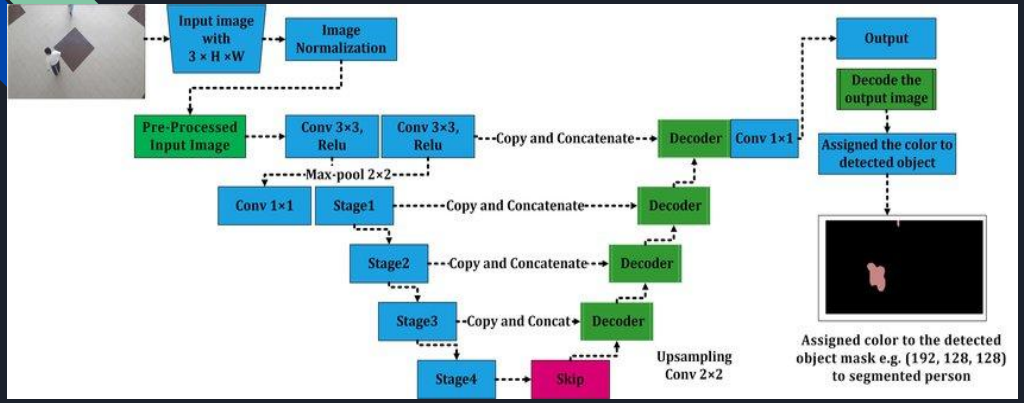


Fig 7.1. ResNet101 U-Net Architecture (Ahmed et al., 2020)

Model	ResNet101 U-Net
Model Size (MB)	196.86
Total Number of Parameters	51,606,191
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	88.11%
Peak Validation_Jaccard_Coeff	0.7531
mIOU	0.5977

Table 7.1. ResNet101 U-Net Performance Metrics

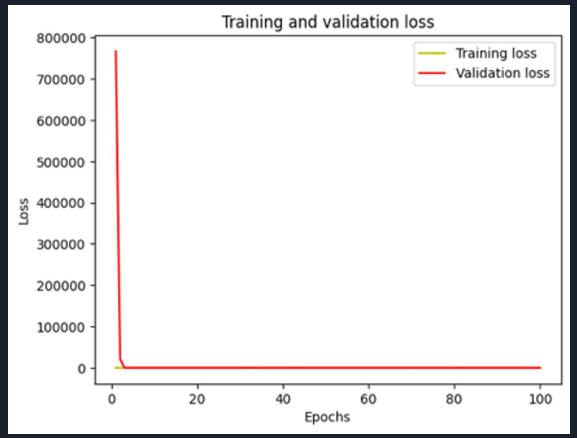


Fig 7.2. ResNet101 U-Net Training Vs Validation Loss Graph

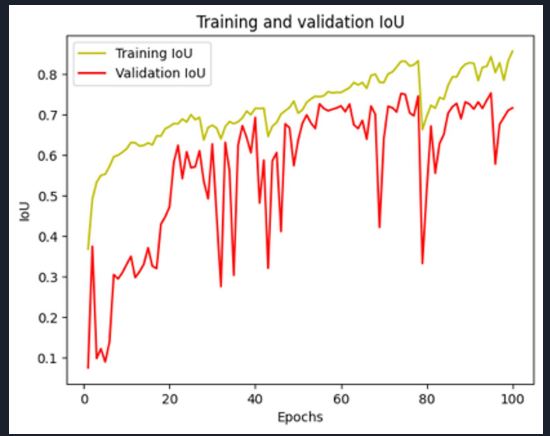


Fig 7.3. ResNet101 U-Net Training Vs Validation IoU Graph

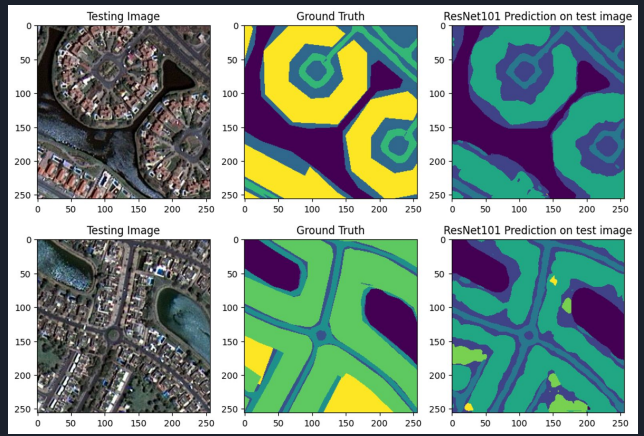


Fig 7.4. ResNet101 U-Net Model Predicted Images

8) ResNet152 U-Net

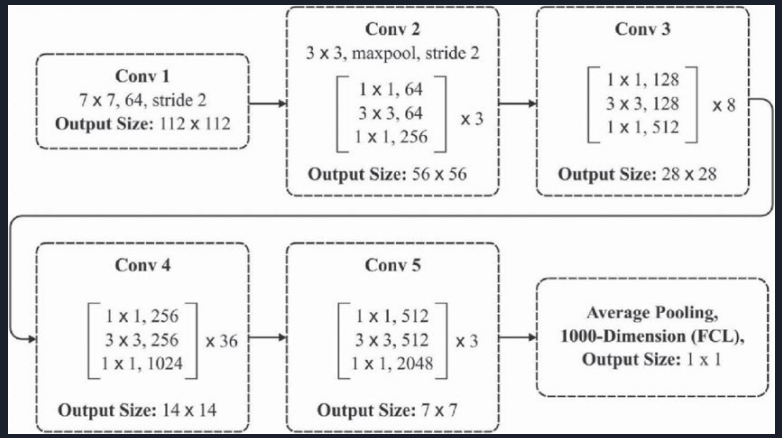


Fig 8.1. ResNet152 U-Net Architecture

Model	ResNet152 U-Net
Model Size (MB)	256.71
Total Number of Parameters	67,295,919
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	88.14%
Peak Validation_Jaccard_Coeff	0.7444
mIOU	0.63

Table 8.1. ResNet152 U-Net Performance Metrics

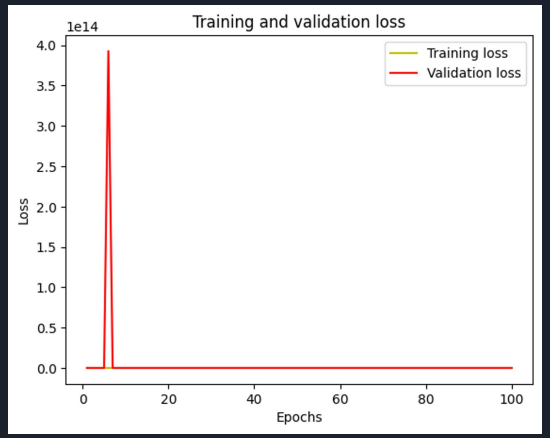


Fig 8.2. ResNet152 U-Net Training Vs Validation Loss Graph

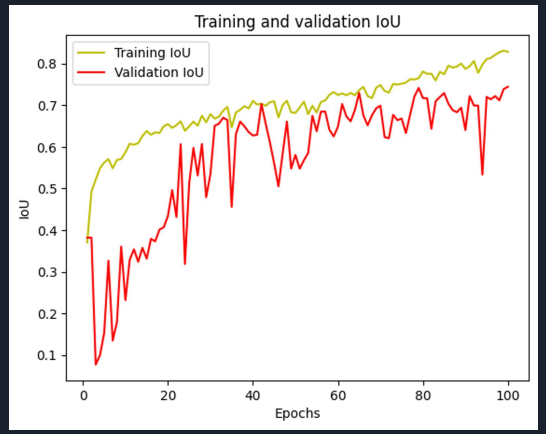


Fig 8.3. ResNet152 U-Net Training Vs Validation IoU Graph

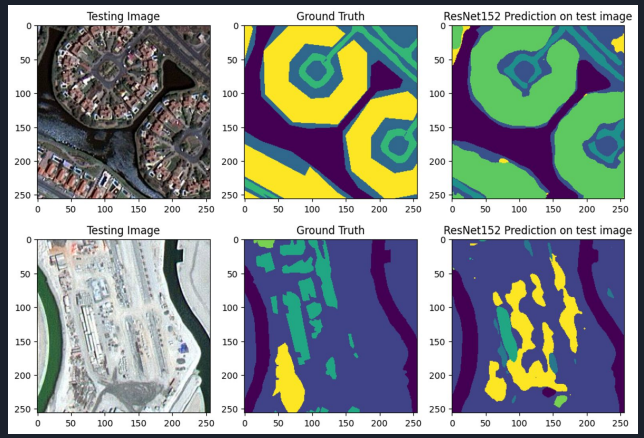


Fig 8.4. ResNet152 U-Net Model Predicted Images

ResNet Predictions Visual Comparison

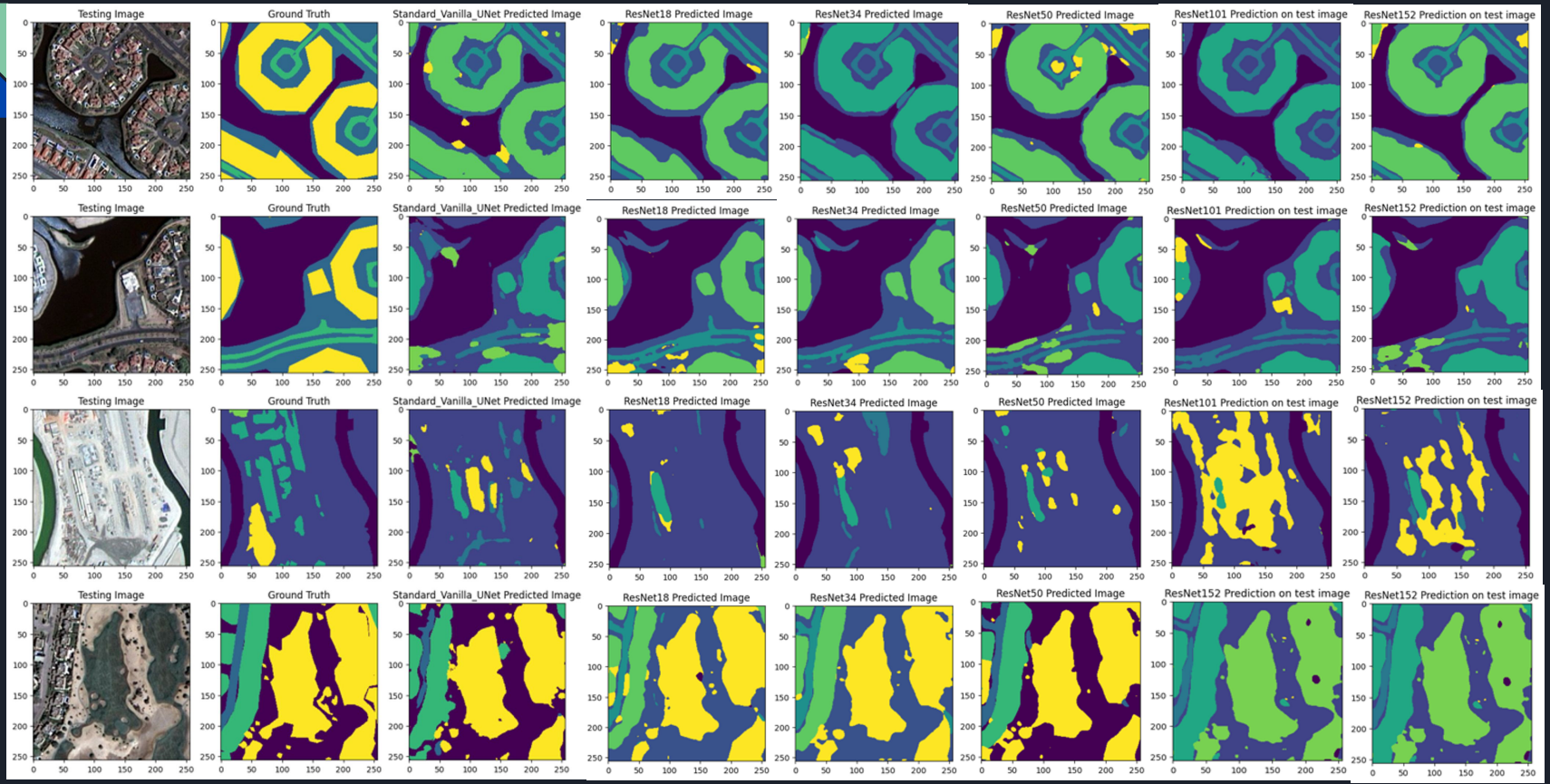


Fig 8. 5. ResNet U-Net Predictions Visual Comparison (ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152)

9) DeepLabV3+ U-Net

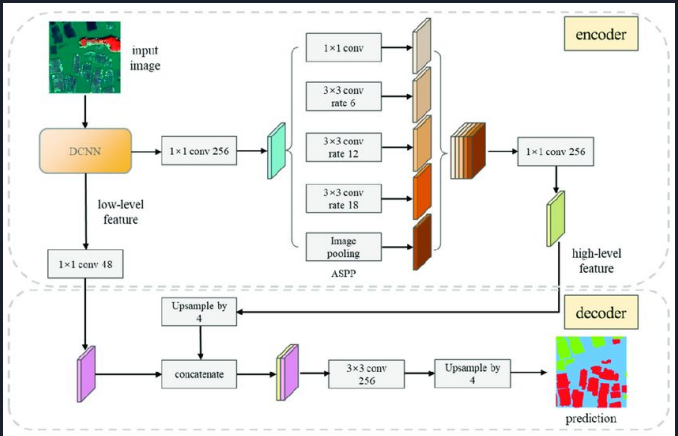


Fig 9.1. DeepLabV3+ Architecture (Chen et al., 2022)

Model	DeepLabV3+ U-Net
Model Size (MB)	68.02
Total Number of Parameters	17,831,494
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	89.03%
Peak Validation_Jaccard_Coeff	0.7981
mIOU	0.6496

Table 9.1. DeepLabV3+ U-Net Performance Metrics

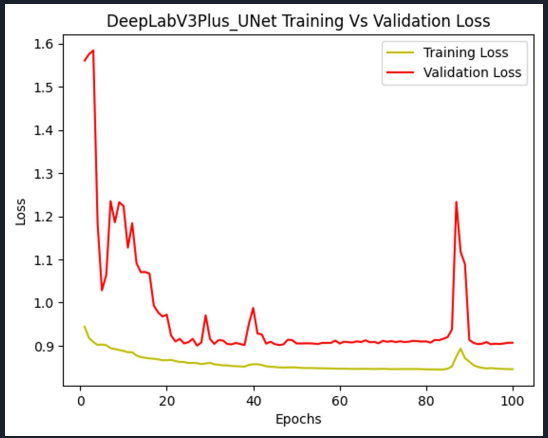


Fig 9.2. DeepLabV3+ U-Net Training Vs Validation Loss Graph

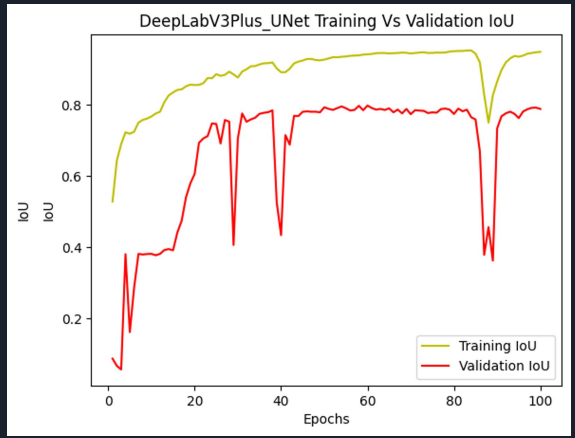


Fig 9.3. DeepLabV3+ U-Net Training Vs Validation IoU Graph

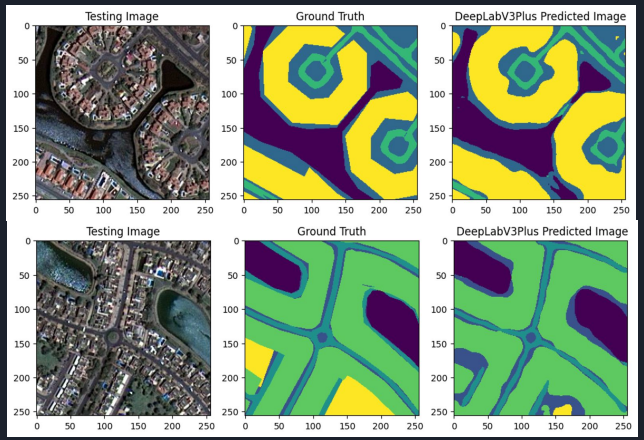


Fig 9.4. DeepLabV3+ U-Net Model Predicted Images

10) DenseNet121 U-Net

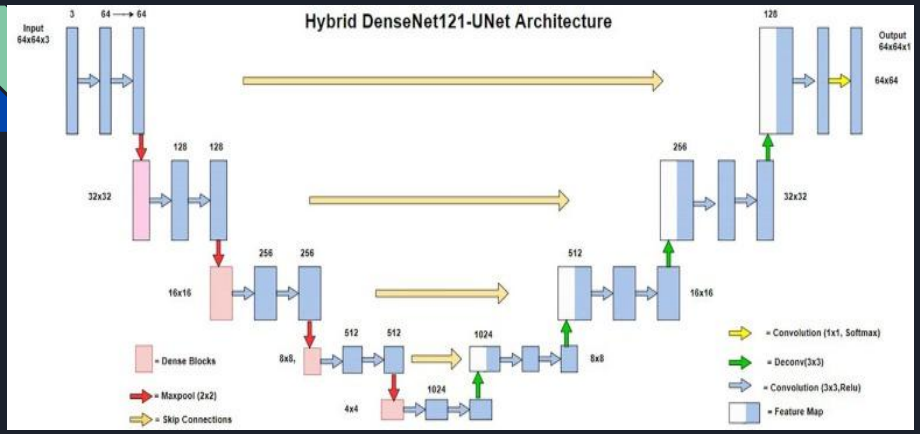


Fig 10.1. DenseNet121 U-Net Architecture (Cinar et al., 2022)

Model	DenseNet121 U-Net
Model Size (MB)	62.60
Total Number of Parameters	16,410,566
Batch_Size	16
Epochs	100
Peak Validation_Accuracy	89.15%
Peak Validation_Jaccard_Coeff	0.7999
mIOU	0.5045

Table 10.1. DenseNet121 U-Net Performance Metrics

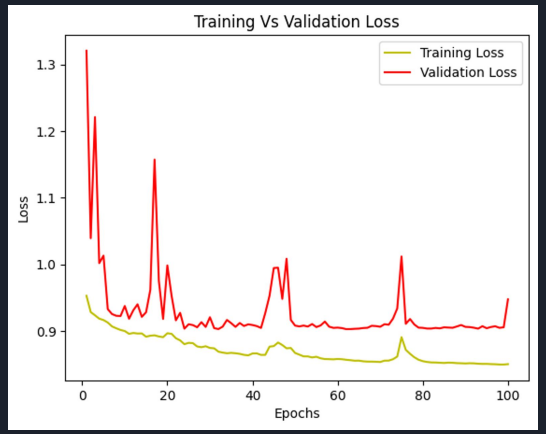


Fig 10.2. DenseNet121 U-Net Training Vs Validation Loss Graph

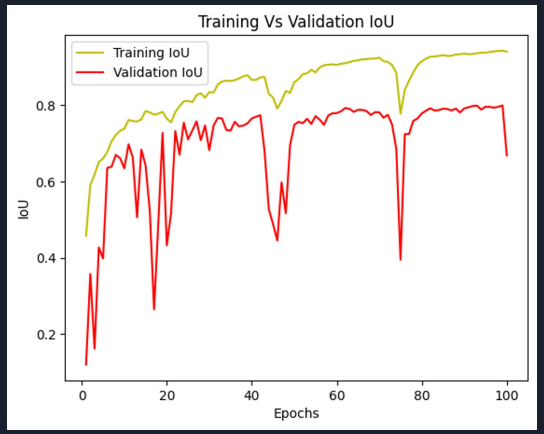


Fig 10.3. DenseNet121 U-Net Training Vs Validation IoU Graph

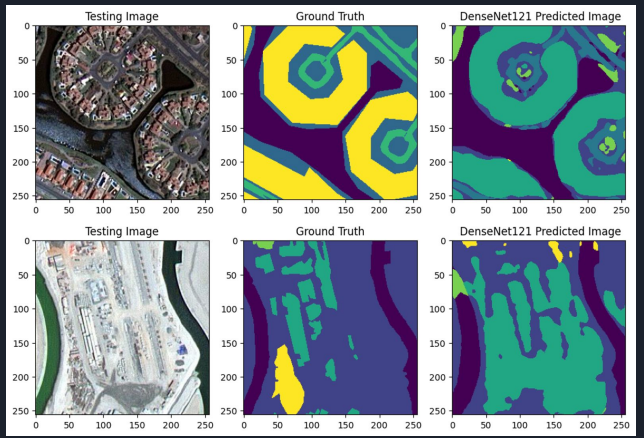
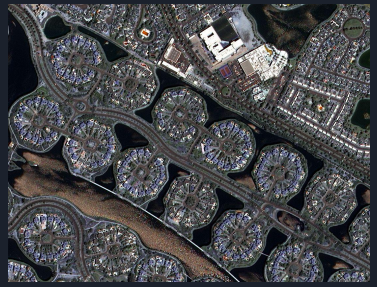


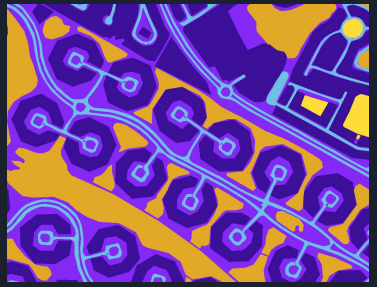
Fig 10.4. DenseNet121 U-Net Model Predicted Images

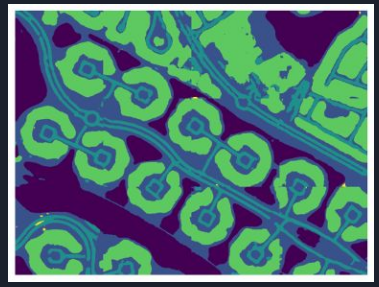
Results Table

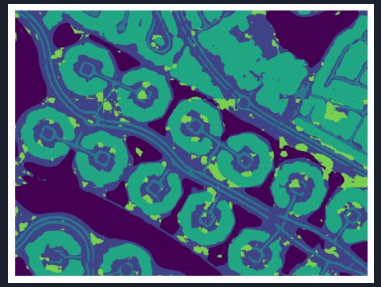
Dataset	Model	Model Size (MB)	Total Number of Parameters	Batch Size	Epochs Trained	Peak Validation Accuracy	Peak Validation Jaccard Coeff	mIoU
MBRSC	Standard Vanilla U-Net	7.41	1,941,190	16	100	85.90%	0.7428	0.5966
MBRSC	VGG16 U-Net	98.66	25,862,662	16	100	88.10%	0.7697	0.6059
MBRSC	VGG19 U-Net	118.91	31,172,358	16	100	88.03%	0.7443	0.5524
MBRSC	ResNet18 U-Net	54.71	14,341,295	16	100	87.54%,	0.7622	0.603
MBRSC	ResNet34 U-Net	93.3	24,456,879	16	100	87.88%	0.7622	0.63
MBRSC	ResNet50 U-Net	124.21	32,561,839	16	100	87.98%,	0.759	0.619
MBRSC	ResNet101 U-Net	196.86	51,606,191	16	100	88.11%	0.7531	0.5977
MBRSC	ResNet152 U-Net	256.71	67,295,919	16	100	88.14%	0.7444	0.63
MBRSC	DeepLabV3+ U-Net	68.02	17,831,494	16	100	89.03%	0.7981	0.6496
MBRSC	DenseNet121 U-Net	62.6	16,410,566	16	100	89.15%	0.7999	0.5045

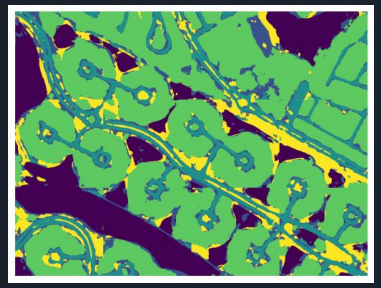
All Models Visual Comparison(Unpatched Predicted Images)

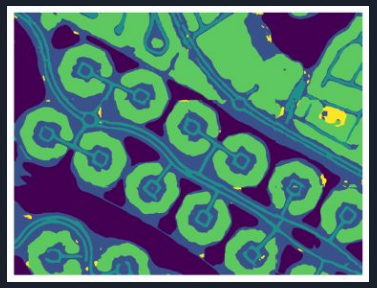

Original Image

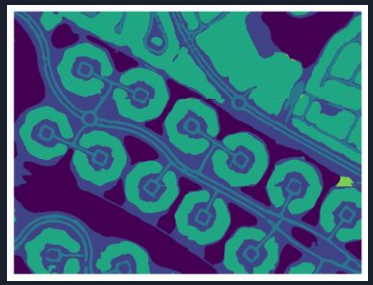

Ground Truth

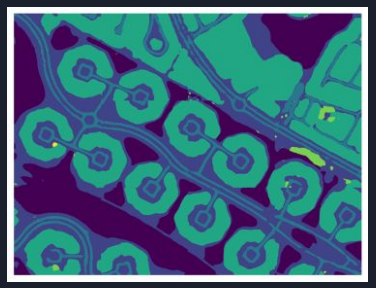

Standard U-Net

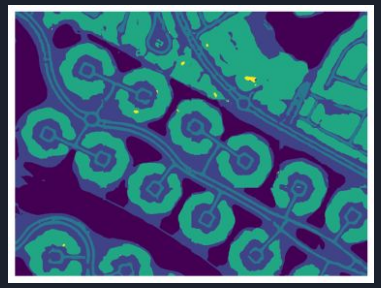

VGG16 U-Net

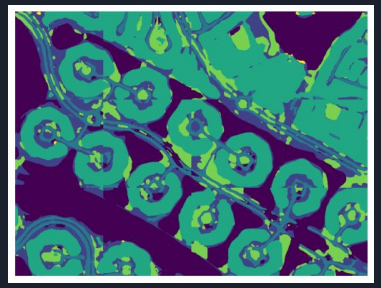

VGG19 U-Net

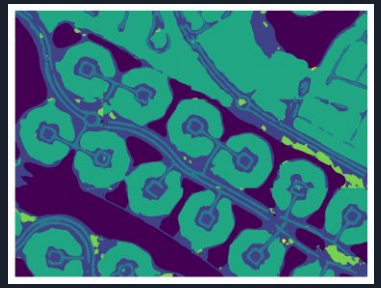

ResNet18 U-Net

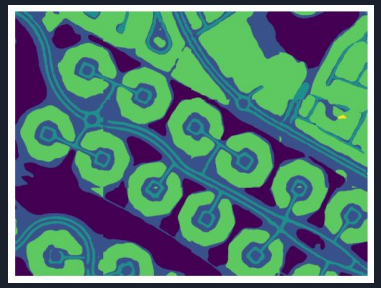

ResNet34 U-Net


ResNet50 U-Net


ResNet101 U-Net


DenseNet121 U-Net


ResNet152 U-Net


DeepLabV3Plus U-Net

Conclusion

In conclusion, my investigation into U-Net–based semantic segmentation for aerial (satellite) imagery demonstrates that deeper encoder backbones—particularly ResNet, DeepLabV3+ and DenseNet-based U-Nets—consistently achieved higher Intersection over Union (IoU) and overall accuracy compared to shallower architectures, even without employing extensive data augmentation techniques. Standard Vanilla U-Net served as a strong baseline; however, integrating more advanced encoders effectively captured complex spatial features such as buildings, roads, water bodies, and vegetation. These findings underscore the robustness and efficiency of hybrid CNN approaches for for capturing complex spatial features in satellite images, and highlight their capacity to support diverse real-world applications ranging from urban planning to environmental monitoring.

Future Work

- **Advanced Architectures:** Investigate transformer-based or hybrid CNN–transformer models to capture even richer spatial and contextual relationships.
- **Data Augmentation:** Adding synthetic or real-world augmentations could help improve generalization in low-data settings.
- **Multi-scale Feature Extraction:** Explore pyramid pooling, atrous convolutions, or feature pyramids for better segmentation of small or irregular targets.
- **Domain Adaptation:** Extend the model to other remote sensing datasets with varying resolutions, illumination, or seasonal conditions.
- **Edge Deployment:** Optimize and compress models for real-time inference on drones or mobile devices in resource-limited environments.

References

1) Davies, A. J. (2022, March 31). *Semantic segmentation of aerial imagery using U-Net in Python*. Towards Data Science. https://towardsdatascience.com/semantic-segmentation-of-aerial-imagery-using-u-net-in-python-552705238514/
2) Bouguettaya, A., Zarzour, H., Kechida, A., & Taberkit, A. (2022). Deep learning techniques to classify agricultural crops through UAV imagery: A review. *Neural Computing and Applications, 34*. https://doi.org/10.1007/s00521-022-07104-9
3) Rahnemoonfar, M., Chowdhury, T. & Murphy, R. RescueNet: A High Resolution UAV Semantic Segmentation Dataset for Natural Disaster Damage Assessment. *Sci Data* 10, 913 (2023). https://doi.org/10.1038/s41597-023-02799-4
4) Humans in the Loop. (n.d.). *Semantic segmentation dataset*. Humans in the Loop. Retrieved February 18, 2025, from https://humansintheloop.org/resources/datasets/semantic-segmentation-dataset-2/
5) Islam, N., Hossain, M. F., & Hossain, M. A. (2024). *Semantic segmentation in satellite imagery: An attentive U-Net approach*. In 2024 IEEE 2nd International Conference on Information and Communication Technology (ICICT) (pp. 259-263). IEEE. https://doi.org/10.1109/ICICT64387.2024.10839725
6) Tong, J., Gao, F., Liu, H., Huang, J., Liu, G., Zhang, H., & Duan, Q. (2023). A study on identification of urban waterlogging risk factors based on satellite image semantic segmentation and XGBoost. *Sustainability, 15*(8), 6434. https://doi.org/10.3390/su15086434
7) Sugirtha, T., & Sridevi, M. (2022). *Semantic segmentation using modified U-Net for autonomous driving*. In 2022 IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-6). IEEE. https://doi.org/10.1109/IEMTRONICS55184.2022.9795710
8) Rahman, T., & Mahanta, L. (2024). Evaluating the deep learning models performance for segmentation of oral epithelial dysplasia: A histological data-driven approach. *Prabha Materials Science Letters, 3*, Article 7. https://doi.org/10.33889/PMSL.2024.3.1.007
9) Kanaeva, I., & Ivanova, J. (2021). Road pavement crack detection using deep learning with synthetic data. *IOP Conference Series: Materials Science and Engineering, 1019*(1), 012036. https://doi.org/10.1088/1757-899X/1019/1/012036
10) Daniel, J., Rose, J. T. A., Vinnarasi, F. S. F., & Rajinikanth, V. (2022). VGG-UNet/VGG-SegNet supported automatic segmentation of endoplasmic reticulum network in fluorescence microscopy images. *Scanning, 2022*, Article 7733860. https://doi.org/10.1155/2022/7733860
11) Chen, Y. (2023). Application of ResNet18-Unet in separating tumors from brain MRI images. *Journal of Physics: Conference Series, 2580*(1), 012057. https://doi.org/10.1088/1742-6596/2580/1/012057
12) Manos, E., Witharana, C., Udawalpola, M., Hasan, A., & Liljedahl, A. (2022). Convolutional neural networks for automated built infrastructure detection in the Arctic using sub-meter spatial resolution satellite imagery. *Remote Sensing, 14*(11), 2719. https://doi.org/10.3390/rs14112719
13) Ahmed, I., Ahmad, M., Khan, F., & Asif, M. (2020). Comparison of deep-learning-based segmentation models: Using top view person images. *IEEE Access, PP*, 1–1. https://doi.org/10.1109/ACCESS.2020.3011406
14) Pustokhin, D., Pustokhina, I., Dinh, P., Phan, S., Nhu, N., Joshi, G. P., & Shankar, K. (2020). An effective deep residual network-based class attention layer with bidirectional LSTM for diagnosis and classification of COVID-19. *Journal of Applied Statistics, 50*, 1–18. https://doi.org/10.1080/02664763.2020.1849057
15) Chen, Y., He, G., Yin, R., Zheng, K., & Wang, G. (2022). Comparative study of marine ranching recognition in multi-temporal high-resolution remote sensing images based on DeepLab-v3+ and U-Net. *Remote Sensing, 14*(22), 5654. https://doi.org/10.3390/rs14225654
16) Cinar, N., Ozcan, A., & Kaya, M. (2022). A hybrid DenseNet121-UNet model for brain tumor segmentation from MR images. *Biomedical Signal Processing and Control, 76*, 103647. https://doi.org/10.1016/j.bspc.2022.103647

THANK YOU