OBJECT DETECTION IN SATELLITE IMAGES

Presented by –
Sumit Gaurav, 2402102012,
M.Tech - CSP

EE644 – Image Processing

TABLE OF CONTENTS

INTRODUCTION

Multi-Scale Object Detection Models

Problem Statement –

Multi-scale object detection (MOD) in satellite imagery remains a significant challenge due to the vast variations in object sizes and distributions.

A key limitation of YOLT is its single-network architecture, which applies a uniform feature extraction approach to objects of all scales, leading to suboptimal performance.

To address these challenges, we propose MOD-YOLT architecture:

- **Scale Classification Module**: Categorizes objects into small, medium, and large scales to apply targeted detection strategies.
- **Multi-Scale Training Strategies**: Uses different network architectures and training techniques to improve feature extraction for each object scale.
- **Improved Fusion Mechanism**: Merges multi-scale detection results to enhance localization accuracy and classification performance.

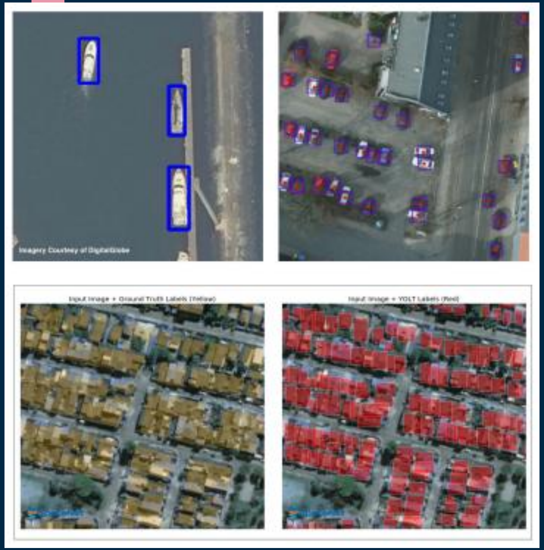


Fig – Example of Object Detection in Satellite images

Research Papers

01

Study of YOLT and MOD-YOLT techniques

DATASET – state-of-the-art dataset

1. AIIA2018_2nd dataset

- Dataset: It is a dataset of satellite remote sensing images, which covers six classes: airport, airplane, harbour, boat, oilcan, bridge.
- The dataset includes 2421 images whose size varies from 512x512 pixels to 2800x2800 pixels.
- Random 450 images are selected as test data, and rest as training data.

Fig – Some Examples from AIIA2018_2nd dataset

2. COWC Dataset

- The dataset encompasses approximately 33,000 unique car instances from six distinct locations.
- The imagery is collected via aerial platforms with a nadir (top-down) view, resembling satellite imagery, and offers a high resolution of 0.15 meters per pixel.
- Each car is annotated with a single pixel marker at its centroid. Notably, only personal vehicles are labeled; commercial vehicles like delivery trucks and tractor-trailers are excluded.
- The COWC dataset is widely used for tasks such as car localization, counting, and detection in overhead imagery.

Fig – COWC dataset images

YOLT ARCHITECTURE

Introduction

YOLT (You Only Look Twice) is an object detection framework optimized for satellite imagery. It improves upon YOLO by handling small, multi-scale objects in large, high-resolution images. YOLT partitions images into smaller cutouts, enhancing detection speed and accuracy.

Features:

- **Multi-Scale Detection**: Uses different input sizes (e.g., 416×416, 544×544) to capture various object scales.
- **Sliding Window Approach**: Overlapping cutouts ensure all regions are analyzed.

Challenges:

- **Fixed Cutout Size**: May miss small objects or fragment large ones.
- **Single Model Limitation**: Struggles to detect objects at vastly different scales.

Layer	Type	Filters	Size/Stride	Output Size
0	Convolutional	32	3×3 / 1	416×416×32
1	Maxpool		2×2 / 2	208×208×32
2	Convolutional	64	3×3 / 1	208×208× 64
3	Maxpool		2×2 / 2	104×104× 64
4	Convolutional	128	3×3 / 1	104×104×128
5	Convolutional	64	1×1 / 1	104×104×64
6	Convolutional	128	3×3 / 1	104×104×128
7	Maxpool		2×2 / 2	52×52×64
8	Convolutional	256	3×3 / 1	52× 52×256
9	Convolutional	128	1×1 / 1	52× 52×128
10	Convolutional	256	3×3 / 1	52× 52×256
11	Maxpool		2×2 / 2	26× 26×256
12	Convolutional	512	3×3 / 1	26× 26×512
13	Convolutional	256	1×1 / 1	26× 26×256
14	Convolutional	512	3×3 / 1	26× 26×512
15	Convolutional	256	1×1 / 1	26× 26×256
16	Convolutional	512	3×3 / 1	26× 26×512
17	Convolutional	1024	3×3 / 1	26× 26×1024
18	Convolutional	1024	3×3 / 1	26× 26×1024
19	Passthrough		10 → 20	26× 26×1024
20	Convolutional	1024	3×3 / 1	26×26×1024
21	Convolutional	N_f	1×1 / 1	26×26×N_f

Table – YOLT Network Architecture

YOLT ARCHITECTURE

Network Architecture for Dense Object

22-layer network downsamples 416×416 inputs to 26×26 grids for dense objects.

Optimized for small, packed objects like cars or buildings, inspired by YOLO.

Passthrough layer adds finer-grained features for better small object detection.

Uses batch norm, leaky ReLU, and final layer predicts bounding boxes and classes.

Testing Procedure

Test images are split into manageable cutouts using a sliding window with 15% overlap.

Cutouts are named by position (e.g., panama50cm|1370 1180 416 416.tif) for tracking.

Each cutout is processed through the trained model for object detection.

Post-Processing for Large-Scale Outputs

Detections from hundreds of cutouts are stitched into a single large image.

Bounding box positions are adjusted to their global coordinates in the original image.

Non-maximal suppression (NMS) is applied to remove overlapping detections from cutout boundaries.

Optimization for Large-Scale Mapping

The 15% overlap ensures full coverage but requires NMS to refine results.

Final outputs provide a comprehensive, large-scale view of detected objects.

This approach maximizes the utility of satellite imagery for mapping and analysis.

Results & Conclusion

Result

- **Universal Classifier Challenges:** A single classifier for all object categories (vehicles and infrastructure) led to spurious detections, especially for airports confused with highways at incorrect scales.
- **Dual Classifier Solution:** Two classifiers were used: one for vehicles/buildings (200m scale) and another for airports (2500m scale). This approach improved accuracy, with detection thresholds of 0.3-0.4 yielding the highest F1 scores.
- **Performance Metrics:** YOLT achieved strong F1 scores: cars (0.90), airplanes (0.87), boats (0.82), buildings (0.61), and airports (0.91). Inference speed was fast (32 km^2/min for most objects, 6000 km^2/min for airports), enabling near real-time processing on large-scale satellite imagery.

Conclusion

- YOLT demonstrates robust multi-scale object detection capabilities, effectively balancing accuracy and speed for diverse satellite imagery applications. Future optimizations can further enhance pre- and post-processing efficiency.

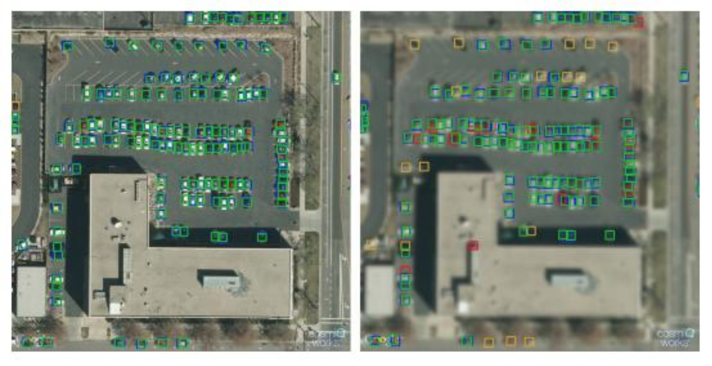


Fig – OD results on different resolutions On left is a 15cm GSD with F1 score 0.94 and on right is a 90 cm GSD with F1 score 0.84.

Object Class	F1 Score	Run Time (km^2/min)
Car†	0.90 ± 0.09	32
Airplane*	0.87 ± 0.08	32
Boat *	0.82 ± 0.07	32
Building*	0.61 ± 0.15	32
Airport*	0.91 ± 0.14	6000

† IOU = 0.25
* IOU = 0.5

Table – YOLT Performance and Speed

MOD YOLT ARCHITECTURE

Introduction

MOD-YOLT builds upon the YOLT framework to enhance object detection in satellite imagery, where objects vary significantly in size and distribution. The approach begins by classifying objects into three categories based on their scale. Each category undergoes tailored training with optimized network architectures and cutout sizes.

Features:

- **Enhanced Multi-Scale Detection**: Objects are categorized into three scales (small, medium, large) to improve detection accuracy.
- **Image Partitioning and Fusion**: Satellite images are divided into various-sized cutouts, and results are stitched together for improved localization.
- **Higher Accuracy**: Outperforms Faster R-CNN and standard YOLT, especially for small-scale objects.

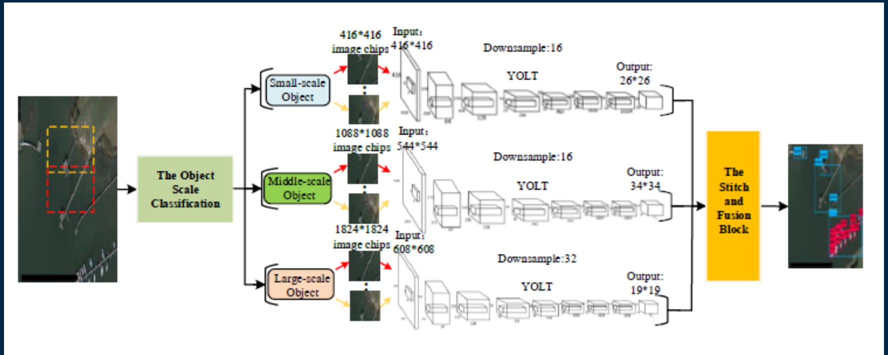


Fig – Overview Framework of MOD-YOLT method for satellite imagery

MOD YOLT ARCHITECTURE

Make reasonable classification for the input images with annotation by the Ratio of the object to the whole image (ROWI).

ROWI Classification

Object Classification

Small Densely Packet Objects - Images are divided into 416×416 cutouts with a 15% overlap using a sliding window approach.

Middle Scale Objects - Images are partitioned into 1088×1088 cutouts with a 15% overlap for better handling.

High Density Scene - Images are split into 1824×1824 cutouts with a 35% overlap to manage complex scene effectively.

Multi YOLT Network

Small Densely Packet Objects - MYN uses a 416×416 input image, downsampled by a factor of 16, to create a 26×26 prediction grid for better detection.

Middle Scale Objects - It processes a 544×544 input image, downsampled by 16, resulting in a 34×34 prediction grid.

High Density Scene - It handles complex scenes with a 608×608 input image, downsampled by 32, producing a 19×19 prediction grid, enhancing feature representation for diverse scales.

Stitch & Fuse Together

Stitching & Fusion- Merge predictions from multiple chips by scaling objects to the full image, selecting the best (class, confidence, coordinates) for each object across MYN's models.

Overlap & Analysis - Use overlaps (15%, 25%, 35%) for full coverage and apply NMS to reduce duplicate detections.

Scale Details - Object classification, image splitting, and input sizes for MYN's models in Fig.

	Scale	Cutouts	Overlap	Input size
$ROWI \leqslant 1/150$	Small-scale Object	416×416	15%	416×416
$1/150 < ROWI \leqslant 1/20$	Middle-scale Object	1088×1088	25%	544×544
$1/20 < ROWI$	Large-scale Object	1824×1824	35%	608×608

Fig – Object scale classification

Results & Conclusion

Result

- **Performance Metrics**: MOD-YOLT achieves the highest true positives (2077) and lowest false negatives (944). It has an F1-score of 0.74179 resp.

- **Average Precision (AP)**: MOD-YOLT excels in detecting small-scale objects (boat, airplane, oilcan) with higher AP. For middle-scale objects (bridge), AP drops slightly, while for large-scale objects (airport, harbour), performance is comparable.

- It achieves a mAP of 78%, 16% and 2% higher than Faster R-CNN and YOLT.

Method	TP	FP	FN	Precision	Recall	F1-score	mAP
Faster R-CNN	1855	429	2672	0.8122	0.4098	0.54471	0.62
YOLT	1589	**113**	1182	**0.9336**	0.5734	0.71048	0.76
MOD-YOLT	**2077**	502	**944**	0.8054	**0.6875**	**0.74179**	**0.78**

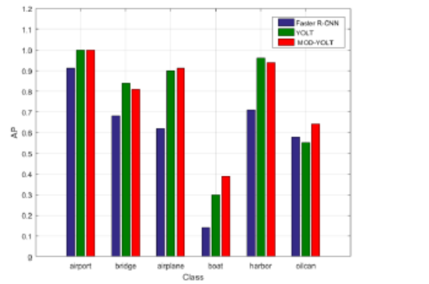


Fig – Detection results of Faster R-CNN, YOLT & MOD-YOLT

Conclusion

- **Improved Performance**: The proposed MOD-YOLT method enhances multi-scale object detection in satellite imagery, achieving 16% and 2% higher mAP than Faster R-CNN and YOLT, respectively.

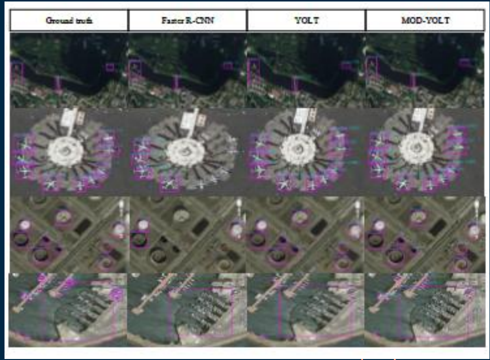


Fig – Visual Detection results on dataset

DL Models

Implementation of
Classification using CNN
and UNET

02

DATASET used

Dataset – Semantic segmentation of aerial imagery

- The dataset consists of aerial imagery of Dubai obtained by MBRSC satellites and annotated with pixel-wise semantic segmentation in 6 classes.
- This dataset contains 945 labelled images which has both images and their masks.
- The images are classified – Water, Land, Vegetation, Road, Buildings, Unlabeled.
- Split the dataset in 85:15 ratio, for training considered 803 data points and for testing 142 images of size 256x256.

Fig – Class labels Semantic seg dataset

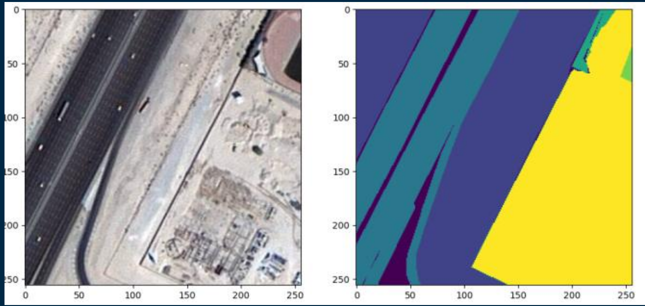


Fig – Image and corresponding masks

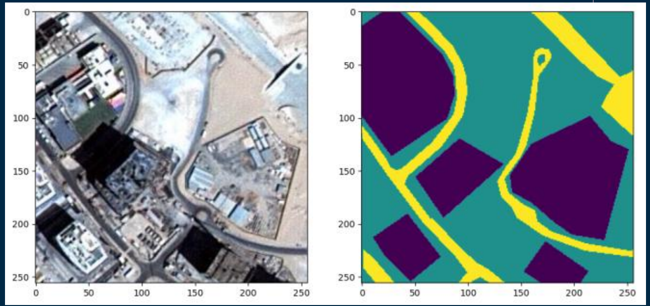
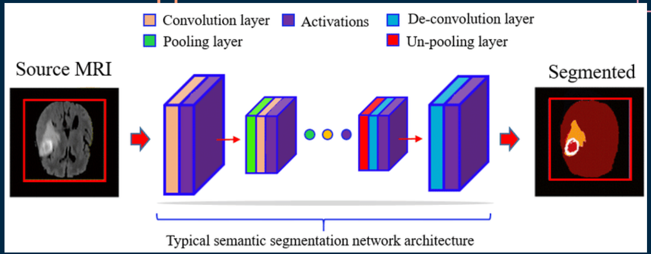


Fig – Image and corresponding masks

CNN ARCHITECTURE & WORKING



Data Pre-processing

Load images and masks from the dataset. Patchify images and masks into smaller patches.

Normalize image data using MinMaxScaler. Convert RGB masks to label masks using rgb_to_label.

Convert label masks to categorical format using to_categorical. Split the dataset into training and testing sets using train_test_split.

CNN for Semantic Segmentation (Encoder)

Define the encoder part of the CNN: Input layer with shape (image_height, image_width, image_channels).

Convolutional layers with Conv2D and MaxPooling2D for downsampling.

Activation function: ReLU. Padding: same.

CNN for Semantic Segmentation (Decoder)

Define the decoder part of the CNN: Convolutional layers with Conv2D and UpSampling2D for upsampling.

Activation function: ReLU. Padding: same.

Output layer with Conv2D and softmax activation for multi-class segmentation.

Training & Result

Compile the model: Optimizer: adam. Loss: Combined DiceLoss and CategoricalFocalLoss.

Metrics: IoU Score and F1 Score. Train the model using model.fit:

Batch size: 16. Epochs: 20. Validation data: (X_test, y_test).

Evaluate the model using model.evaluate: Print test loss, IOU, and F1 score. Print validation loss and accuracy.

Results & Output – CNN

Result

Accuracy	F1 Score	Loss	Validation Accuracy	IoU Score	Validation Loss
0.4856	0.6330	0.4079	0.6296	0.6951	0.5269

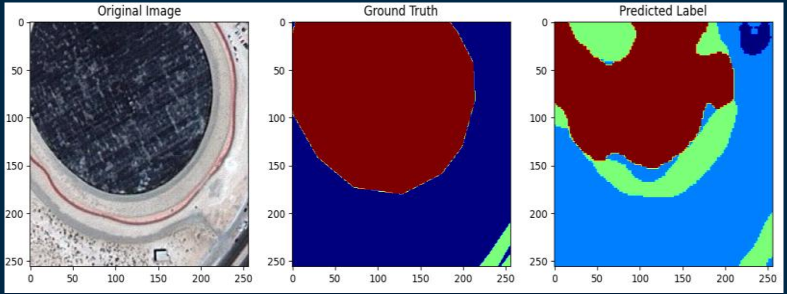


Fig 1 – Test output on random image

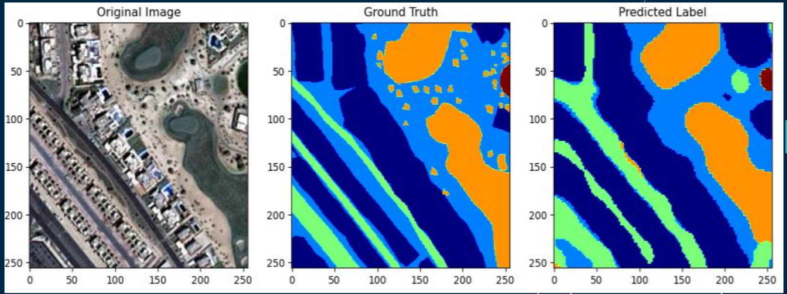
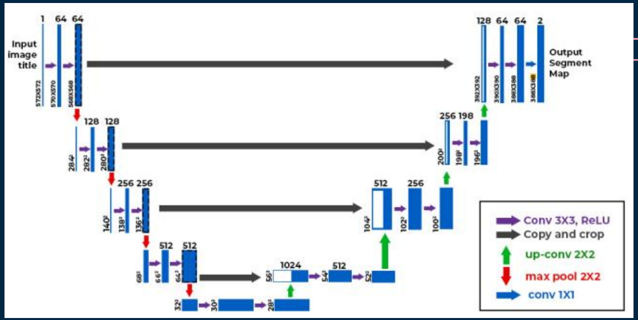


Fig 2 – Test output on random image

UNET ARCHITECTURE & WORKING



Data Pre-processing

Input: Raw image data (X_train, X_test) and corresponding labels (y_train, y_test). Normalize/scale image data.

Resize images to a fixed size (IMG_HEIGHT, IMG_WIDTH). Encode labels into one-hot format (if required). Split data into training and validation sets.

Output: Preprocessed X_train, X_test, y_train, y_test.

UNET for Semantic Segmentation (Encoder)

Input: Preprocessed image data. Define the U-Net model: Contraction Path (Encoder): Convolutional layers with ReLU activation. MaxPooling layers for downsampling. Dropout layers for regularization.

Bottleneck Layer: Convolutional layers with ReLU activation.

UNET for Semantic Segmentation (Decoder)

Expansive Path (Decoder): Transposed Convolution layers for upsampling. Concatenation with corresponding encoder features (skip connections). Convolutional layers with ReLU activation. Output layer with softmax activation for multi-class segmentation.

Compile the model with: Loss function: DiceLoss + FocalLoss.

Training & Result

Input: Compiled U-Net model, preprocessed training data (X_train, y_train), and validation data (X_test, y_test).

Train the model using model.fit(): Predict segmentation masks for test images using model.predict(). Calculate Mean IoU using MeanIoU from Keras. Visualize results: Display a random test image. Display the ground truth label. Display the predicted segmentation mask.

Output: Trained U-Net model, evaluation metrics (Mean IoU), and visualizations.

Results & Output – UNET

Result

Accuracy	Jacard Coefficient	Loss	Validation Accuracy	Validation Jacard Coeff	Validation Loss
0.8775	0.7437	0.8909	0.8399	0.6951	0.9164

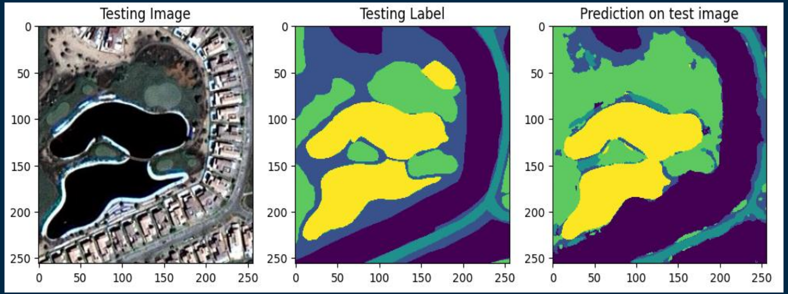


Fig 1 – Test output on random image

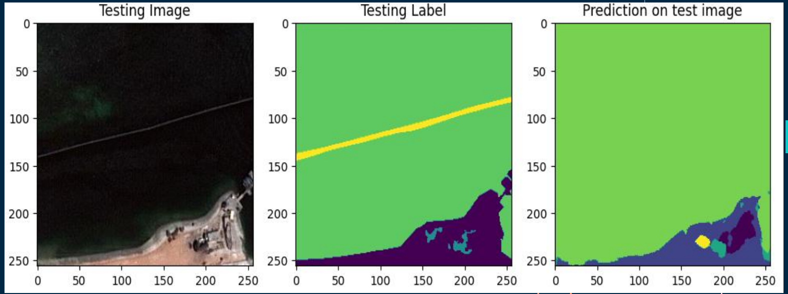


Fig 2 – Test output on random image

Conclusion

Summary and future scope for MOD

03

CONCLUSION

Multi-Scale Object Detection Model -

The MOD-YOLT framework presents a significant improvement in multi-scale object detection for satellite imagery by addressing the limitations of existing methods such as YOLT and Faster R-CNN. Through a novel approach that incorporates scale classification, adaptive training strategies, and an advanced fusion mechanism, MOD-YOLT enhances both localization accuracy and classification performance across varying object sizes.

Takeaways –
Improved Detection Performance: MOD-YOLT outperforms Faster R-CNN and standard YOLT, achieving higher accuracy, especially for small-scale objects.
Multi-Scale Adaptability: The framework effectively handles diverse object sizes by applying different network architectures tailored to specific scales.

After Mid-Sem objectives-

Develop a GUI where a user can upload certain format of image and get object detection.
Try to implement YOLT architecture and achieve comparable results.

REFERENCES

Research Papers -

Multi-Scale Object Detection in Satellite Imagery Based On YOLT

- LINK - https://ieeexplore.ieee.org/document/8898170/figures#figures

You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery

- LINK - https://arxiv.org/pdf/1805.09512v1

You Only Look Twice — Multi-Scale Object Detection in Satellite Imagery With Convolutional Neural Networks

- LINK - https://medium.com/the-downlinq/you-only-look-twice-multi-scale-object-detection-in-satellite-imagery-with-convolutional-neural-38dad1cf7571

Dataset -

Semantic segmentation of aerial imagery

- LINK - https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery/data

THANK YOU

Sumit Gaurav, 2402102012
M.Tech - CSP