Super resolution and Object detection on Remote sensing imagery

CSCI 5561 - Project Presentation

Manpreet Singh
Anwesha Samaddar
Seungho Shin
Shizra Tariq

UNIVERSITY OF MINNESOTA
Driven to Discover℠

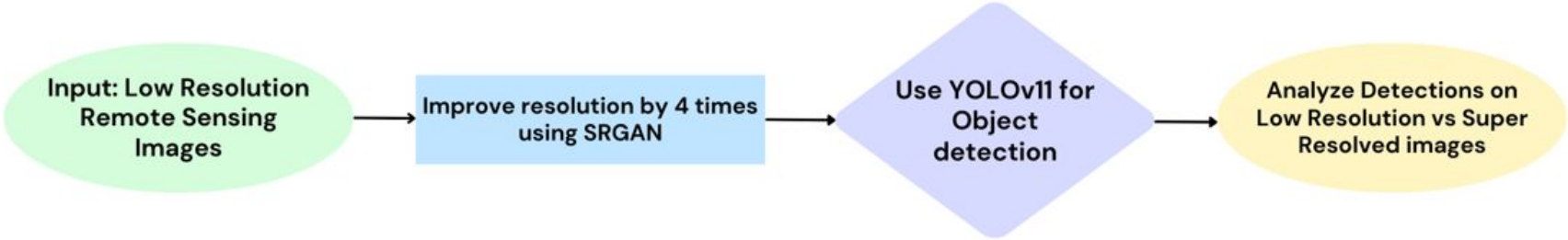
Project Overview

Project Goals:

- Enhance the quality of remote sensing images using advanced ***Super-Resolution*** techniques.
- Improve object detection in aerial imagery using **YOLOv11** and **oriented bounding boxes (OBB).**

Motivation:

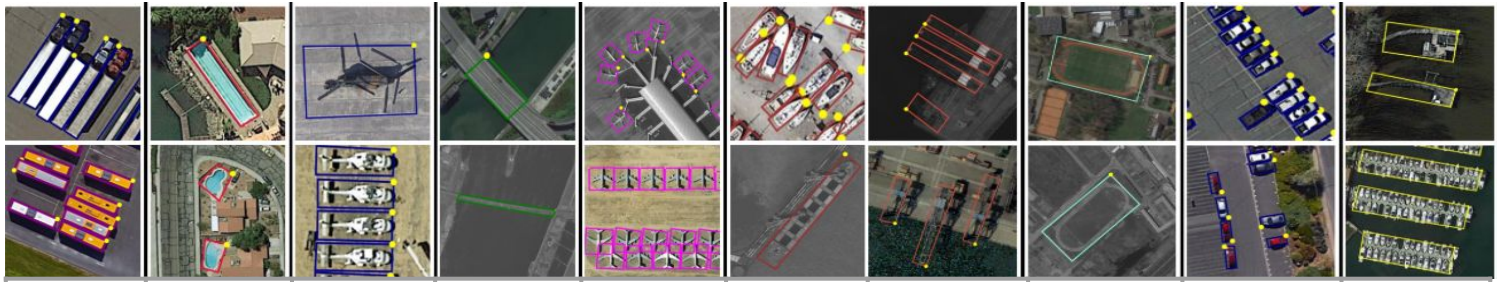
- Practical applications in military surveillance, environmental monitoring, and urban planning.
- Analyzing traffic patterns, parking usage, or urban development with high precision in complex urban environments.



Dataset description

- **DOTA-v1.0**: A large-scale dataset for object detection in aerial images.
  - **Categories**: 15 common categories.
  - **Images**: 2,338 high-resolution images ranging from **800 × 800** to **10000 x 10000** pixels.
  - **Instances**: ~17k annotated objects with arbitrary quadrilateral annotations.
  - **Splits**:
    - Training Set: ~60%
    - Validation Set: ~20%
    - Testing Set: ~20%

Example annotations of DOTA v1

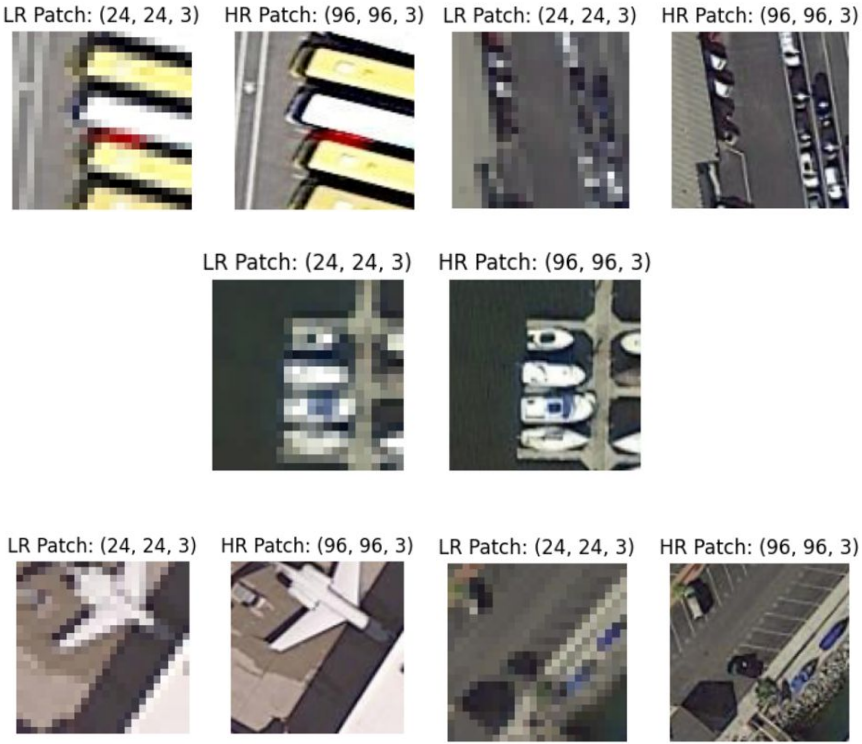


Large-vehicle	Swimming Pool	Helicopter	Bridge	Plane	Ship	Container Crane	Ground track field	Small vehicle	Harbour

Dataset preparation for Super-resolution

Creating pairs of High resolution and Low resolution image patches:

1. **HR Images**: used as targets to train the chosen super-resolution networks
    - Patches: random patches of size **96 × 96 are generated from the training images**
    - Patches normalized to **[-1, 1]**
2. **LR Images**: serve as inputs to the models to simulate real-world low-resolution scenarios, like compressed or blurred images.
    - Generated via **4x bicubic downsampling**
    - Used a Gaussian blur before downsampling to prevent aliasing
    - Normalized images



Paper Implementation: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network: link

Network Overview - GENERATOR:

- **Goal**: Upscale image dimensions by 4x using network: **SRGAN generator (SRResNet)**

1. SRResNet:

A fully convolutional network designed for 4x super-resolution using **residual blocks** and **skip connections**.

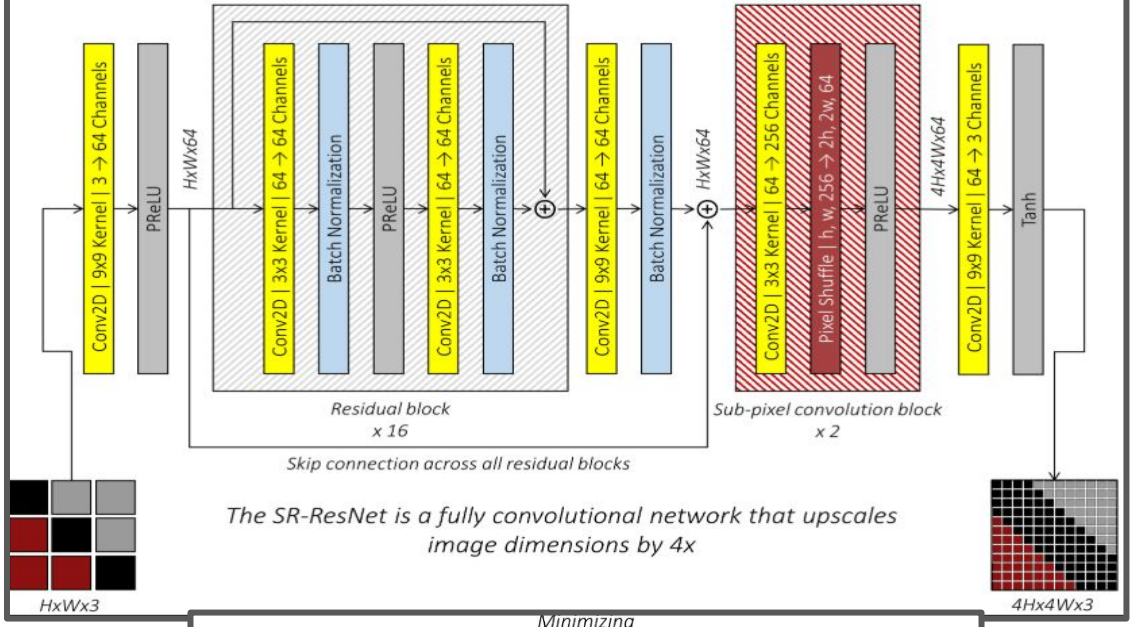
- **Purpose**: Serves as a baseline for **SRGAN** and is trained independently. Used for both comparison and initialization.

SRResNet Architecture:

- **Input**: Low-resolution (LR) image
- **Residual Blocks (16)**: 3×3 convolutions, Batch Norm, PReLU activation, skip connections
- **Upscaling**: 2 subpixel convolutions (×2) for 4x upscale
- **Output**: 4x super-resolved image

Training:

- **Loss Function**: Mean-Squared Error (MSE) between super-resolved image and ground truth HR image.
- **Objective**: Minimize MSE to make super-resolved image closer to original HR.



Minimizing

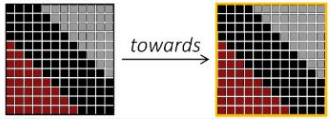
Super-resolved image — Gold high-res image

$$\sum \left(\text{image} - \text{image} \right)^2$$

MSE loss

$$\frac{}{total\ pixels * channels}$$

moves $\theta_{SRResNet}$ in a way that drives



SRGAN: Super-Resolution Generative Adversarial Network

Architecture: Discriminator

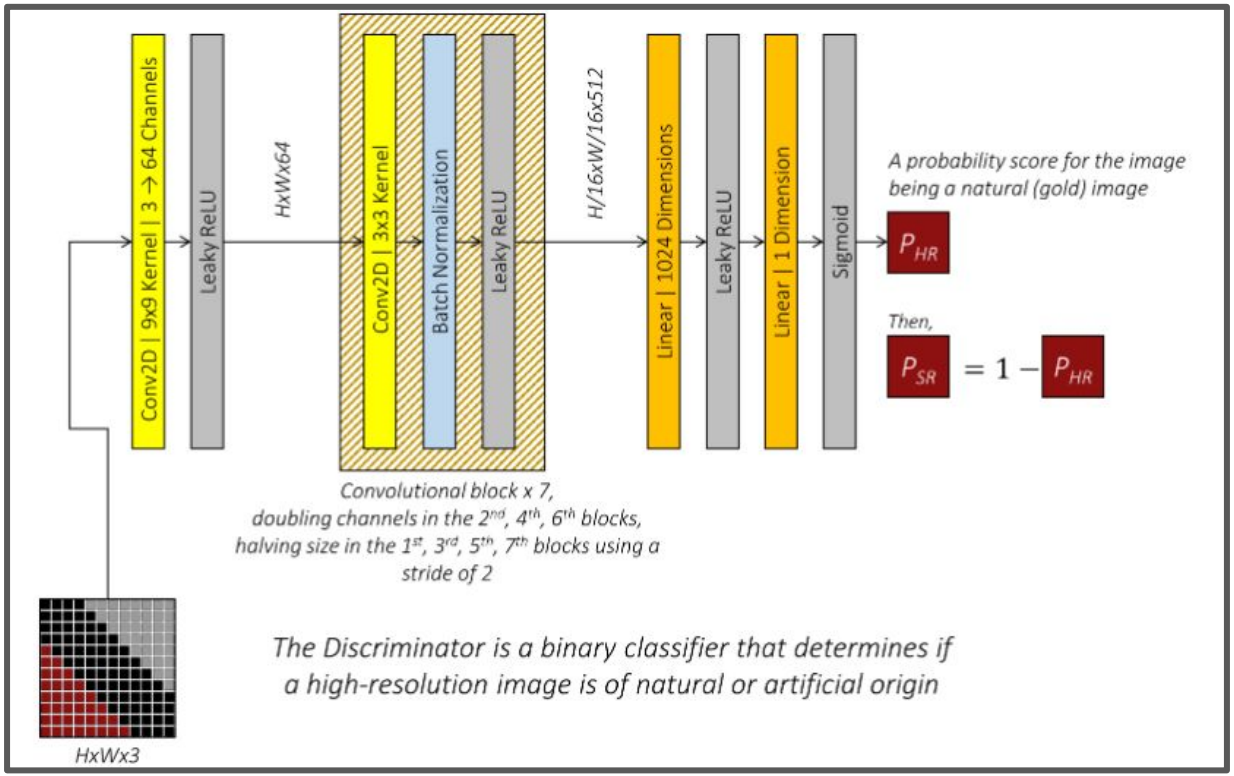
- A convolutional network acting as a binary image classifier.

Discriminator Details:

- **Input**: High-resolution images (real or fake).
- **7 Convolutional Blocks**: 3x3 Conv2d Kernels, batch normalization, leaky ReLU activation
- **Flattened**: Transformed to 1024-size vector, followed by leaky ReLU.
- **Output**: a probability score (using Sigmoid activation) indicating the likelihood of the image being real.

Training:

- **Training**: Generator and Discriminator are alternately trained for short periods.
- **Update**: Each network is updated once before switching to the other.


Conv2D | 9x9 Kernel | 3 → 64 Channels
Leaky ReLU
HxWx64
Conv2D | 3x3 Kernel
Batch Normalization
Leaky ReLU
H/16xW/16x512
Linear | 1024 Dimensions
Leaky ReLU
Linear | 1 Dimension
Sigmoid
A probability score for the image being a natural (gold) image
P_{HR}
Then,
$P_{SR} = 1 - P_{HR}$
Convolutional block x 7, doubling channels in the 2nd, 4th, 6th blocks, halving size in the 1st, 3rd, 5th, 7th blocks using a stride of 2
HxWx3
The Discriminator is a binary classifier that determines if a high-resolution image is of natural or artificial origin

Experiments

Training params	SRResNet Training	SRGAN Training
Epochs	140	300
Training+val images	1869	1869
Time	5 hrs	10 hrs
Pre-trained weights	Trained on MSCOCO '14	Trained on MSCOCO '14
Optimizer	Adam	Adam
Learning Rate	1.00E-04	1.00E-04
GPU	RTX 3050 Ti	RTX 3050 Ti

Results on Test Set Images:

- 469 images of 1024x1024 dimension
- Convert into LR: 256x256
- Generate SR images and compare PSNR/SSIM

Model	Average PSNR	Average SSIM
SRGAN	28.840011	0.785837

Comparing Results with Bicubic and Nearest Neighbour

Input LR image (263,382)

LR image

Low-Resolution (Zoomed-In)

Nearest neighbor

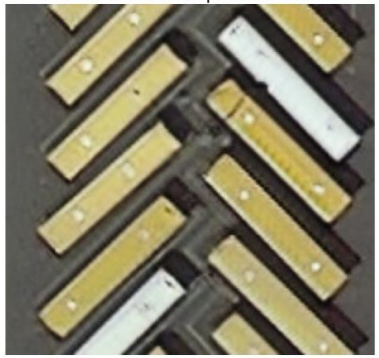
PSNR: 17.35 dB | SSIM: 0.559

Bicubic

PSNR: 18.63 dB | SSIM: 0.594

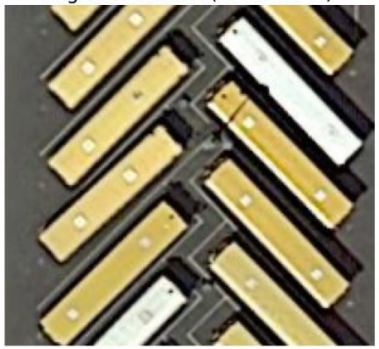
SRGAN

PSNR: 18.64 dB | SSIM: 0.690



Ground Truth HR

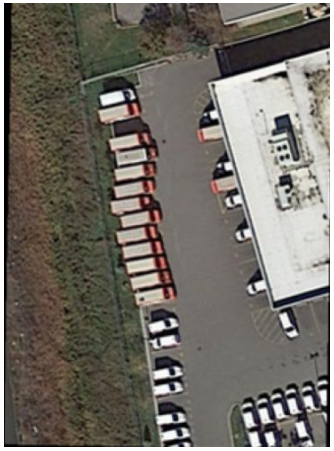
High-Resolution (Zoomed-In)



Comparing Results with Bicubic and Nearest Neighbour

LR image

GT HR



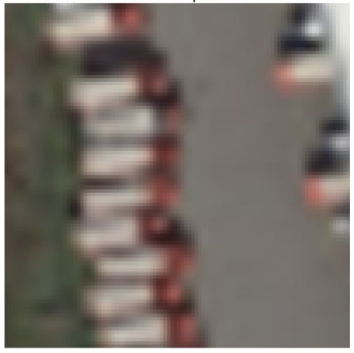
LR image

Nearest neighbor

PSNR: 17.21 dB | SSIM: 0.555

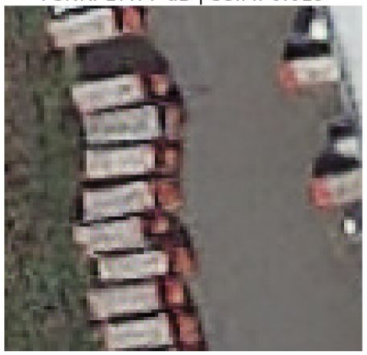
Bicubic

PSNR: 17.44 dB | SSIM: 0.550



SRGAN

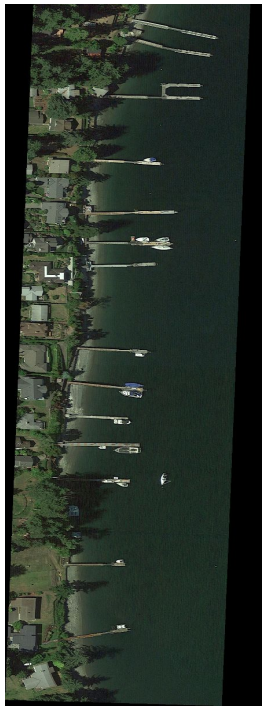
PSNR: 17.77 dB | SSIM: 0.619



GT HR

Object detection - Dataset preparation

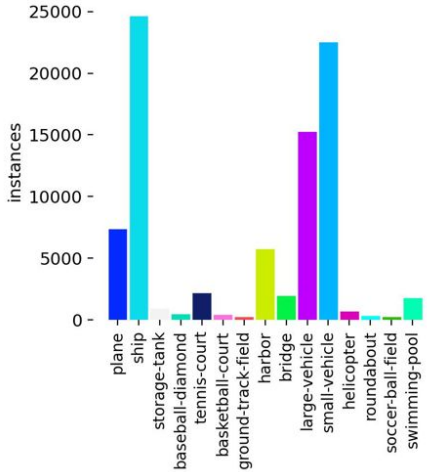
- Original dataset has large image dimension e.g. 4000 x 5000, 1200 x 6000.
- All images are irregular size with some having black regions.



Object detection - Dataset preparation

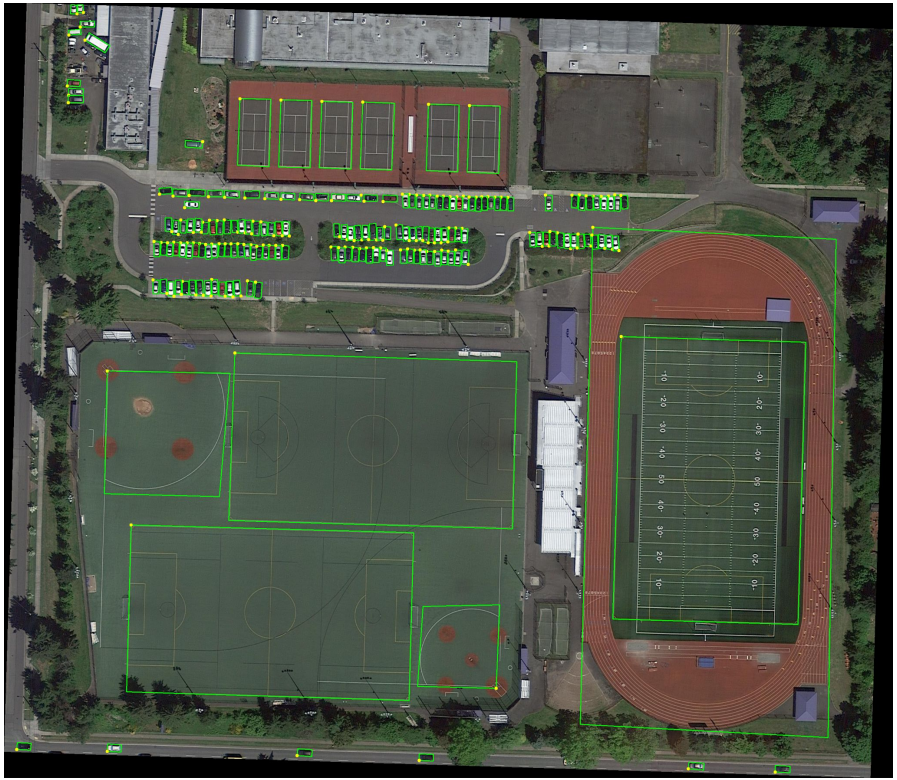
- Original dataset has large image dimension e.g. 4000 x 5000, 1200 x 6000.
- All images are irregular size with some having black regions.

- Original dataset has 15 classes.
- But the distribution of classes in dataset is uneven.

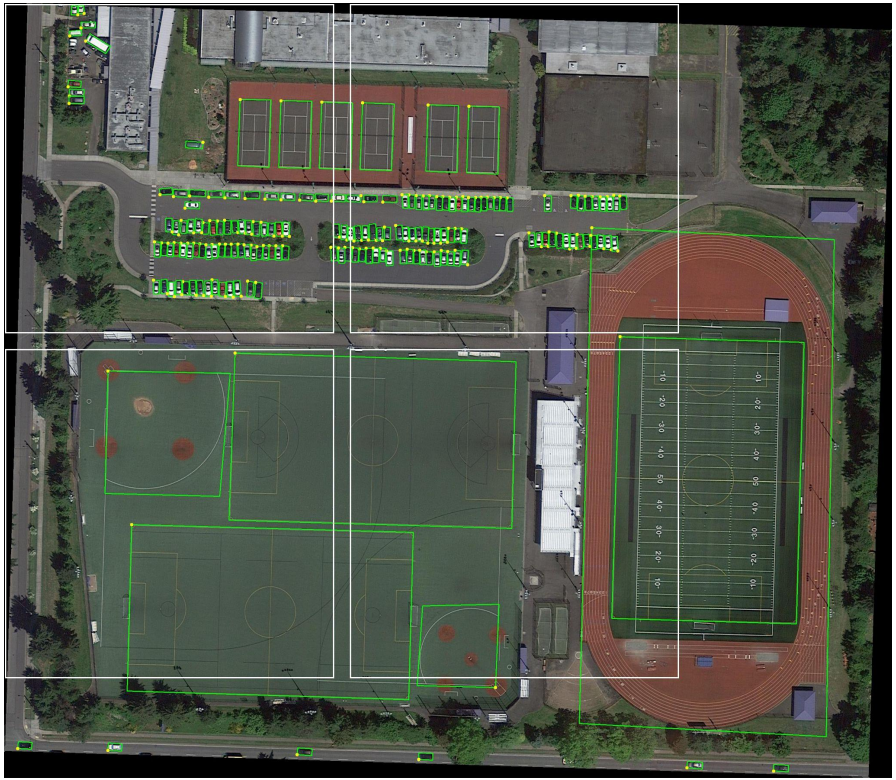


- We have decreased the number of classes to **8**.
- Classes which have **low occurrence** and **large size** are removed.
- E.g. Tennis court, baseball diamond.
- We also create cropped regions of size **1024 by 1024** for training YOLO models.

Original images with their labels

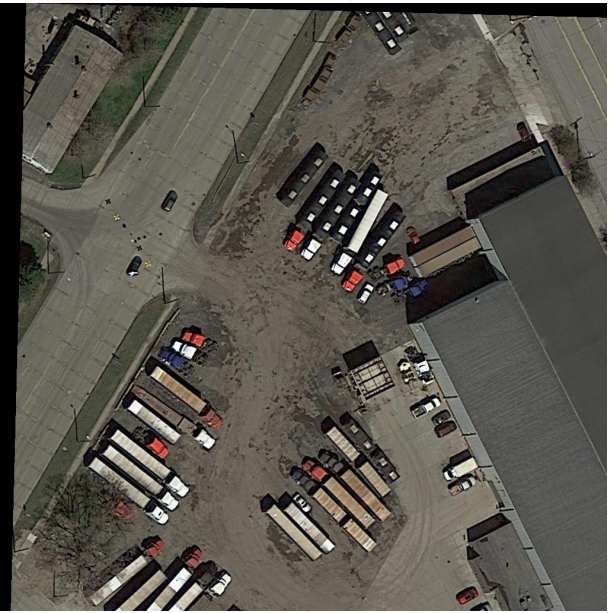


Original images with reduced labels 1024 by 1024 cropped areas

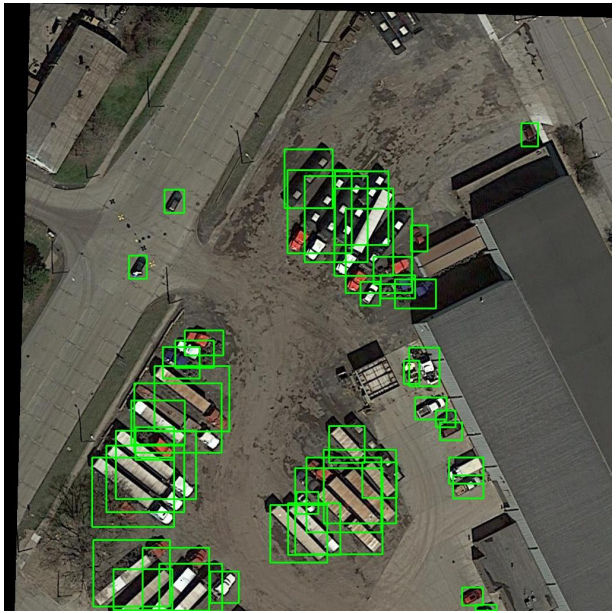


Training - YOLO and YOLO-OBB

Input images



YOLO labels

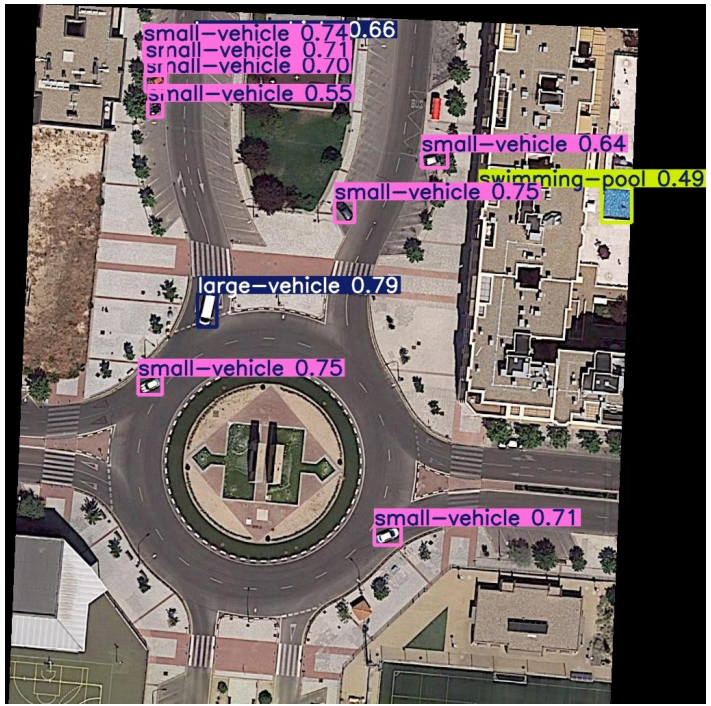


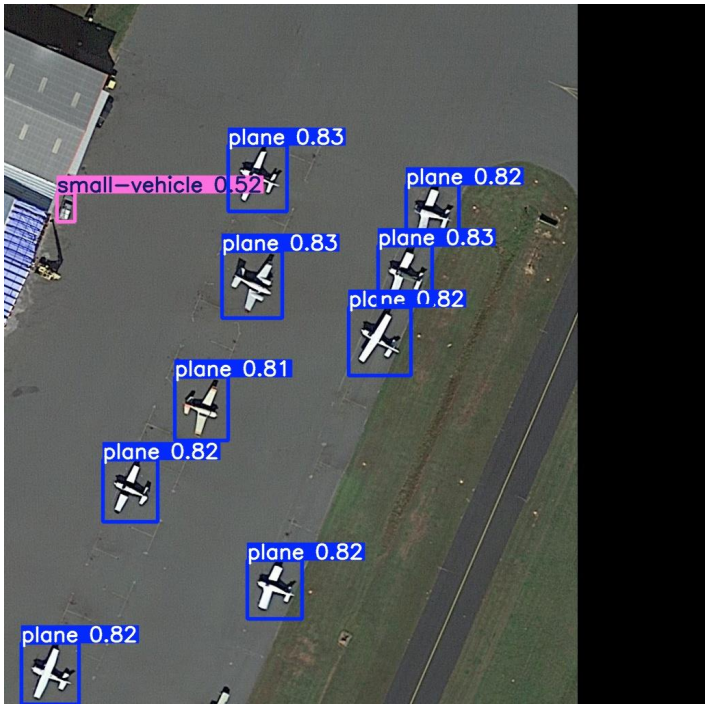
YOLO-OBB labels

- Original label format: x1, y1, x2, y2, x3, y3, x4, y4, category, difficult
- YOLO label format: class_index, x_center, y_center, width, height
- YOLO-OBB labels format: class_index, x1, y1, x2, y2, x3, y3, x4, y4

YOLO training

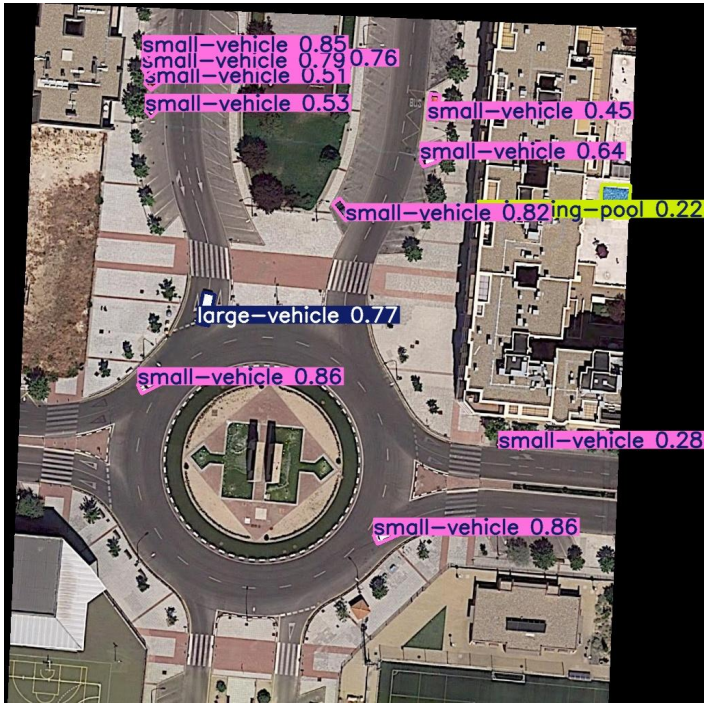
- Train on ~3400 images for simple object detection for 100 epochs.
- Trained YOLO model with **8 classes.**

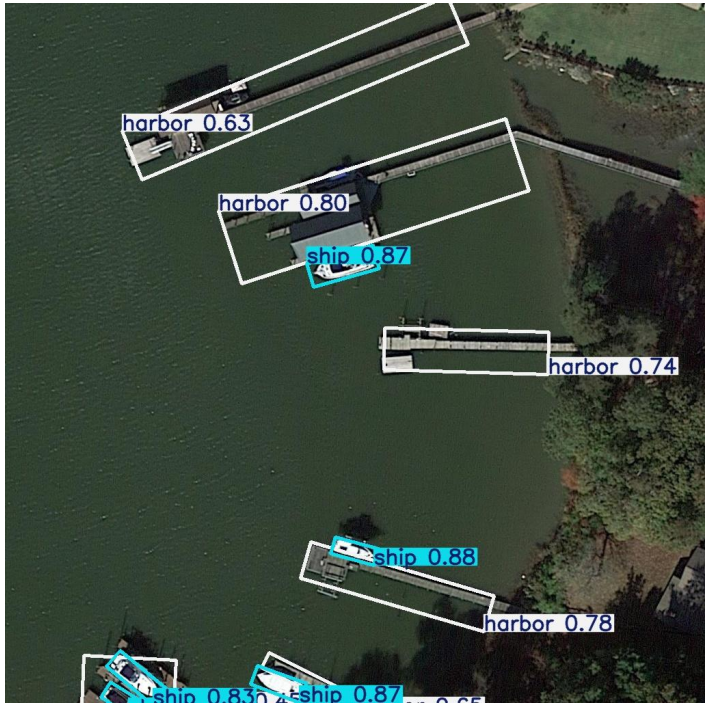




YOLO-OBB training

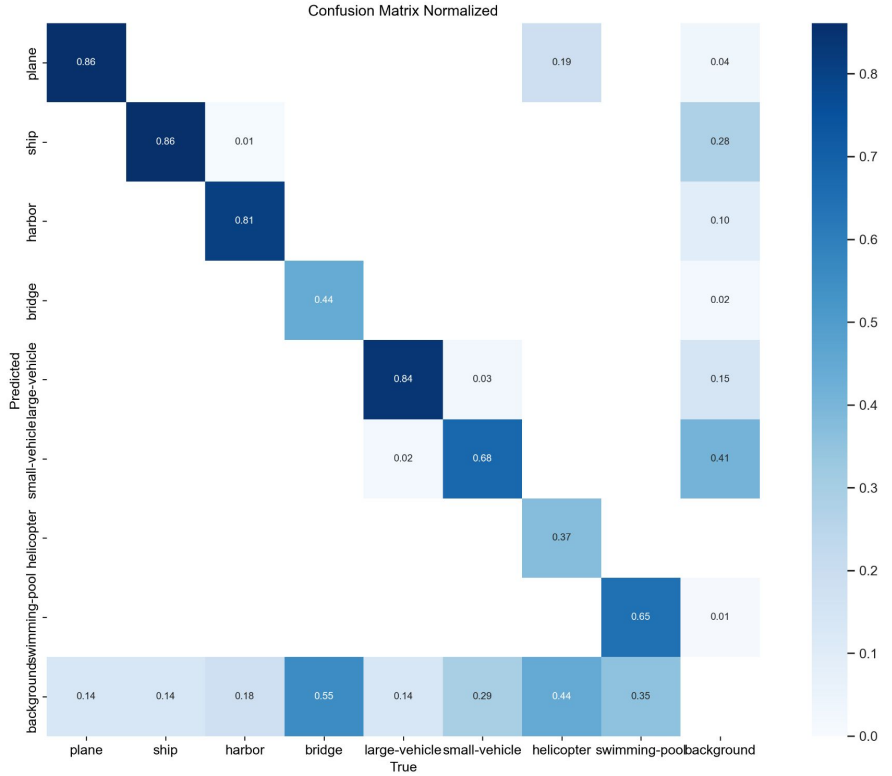
- Train on ~5000 images for oriented object detection for 70 epochs.
- Trained YOLO model with **8 classes.**



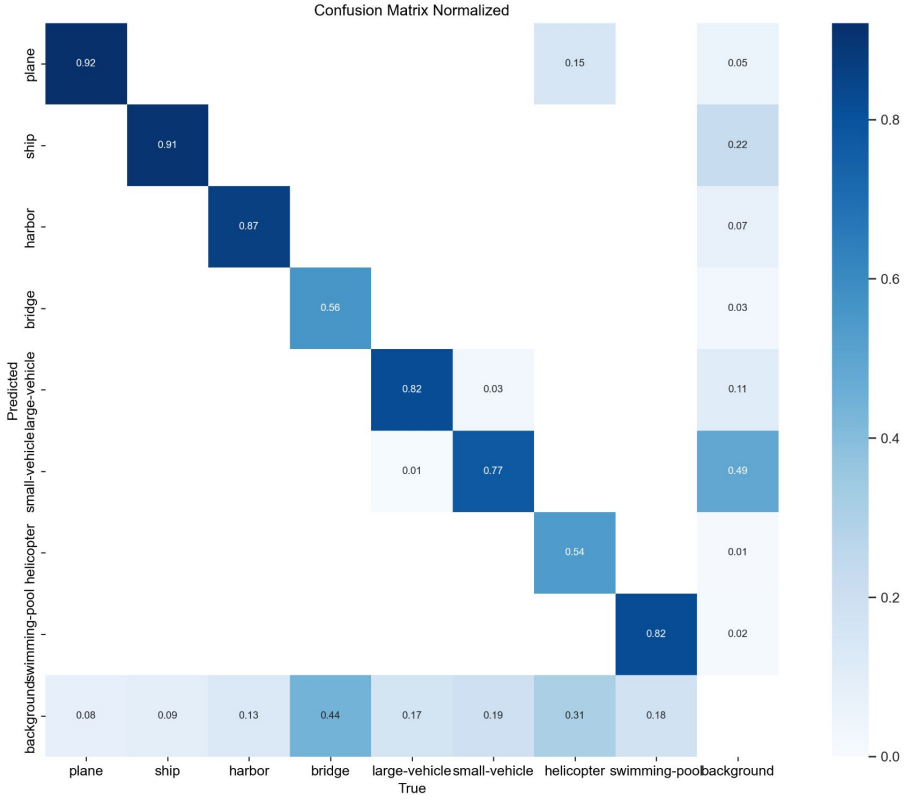


Performance

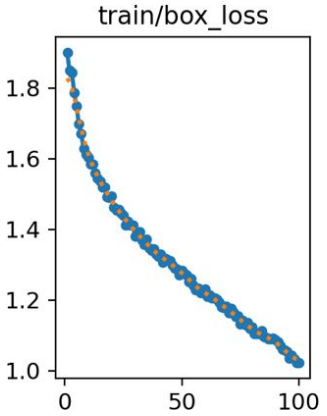
YOLO

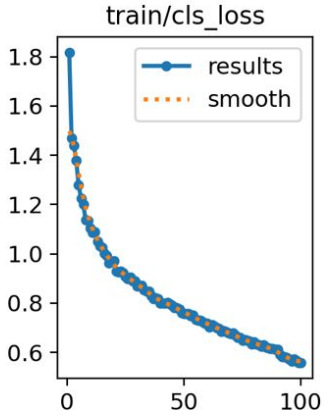


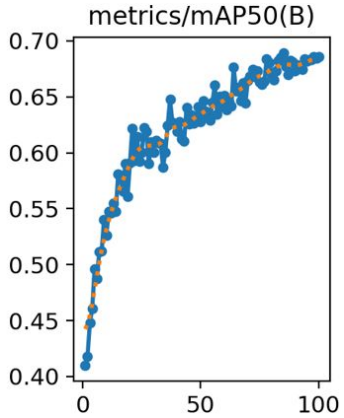
YOLO-OBB

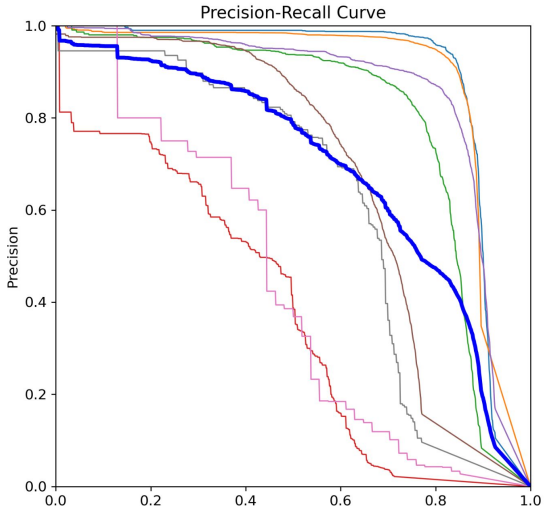


YOLO

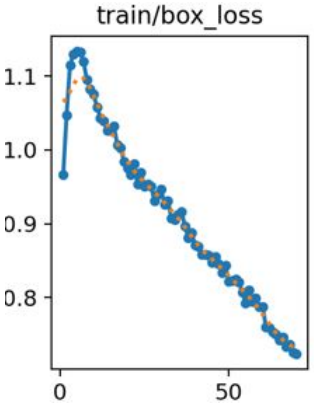


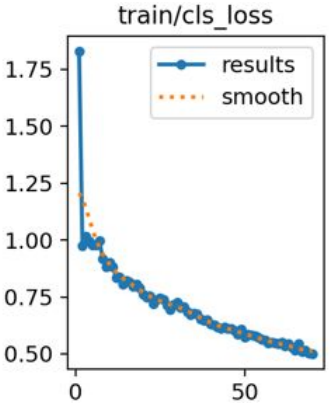


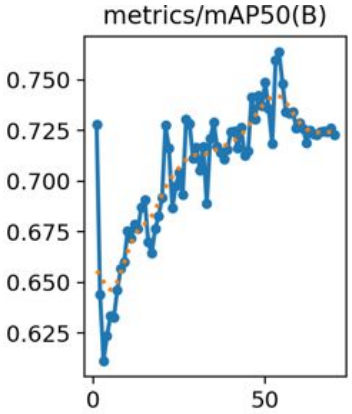


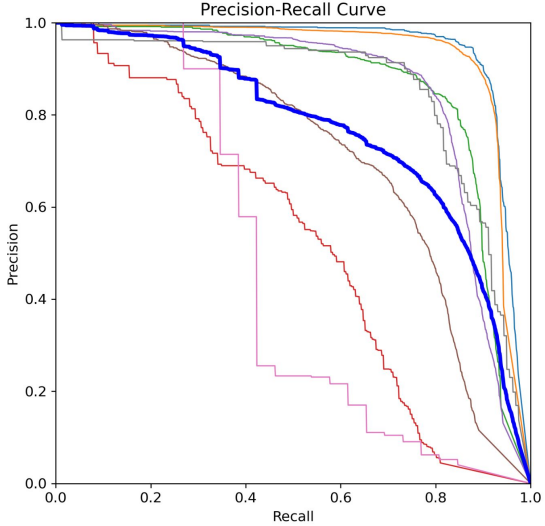


YOLO-OBB









Demo

Members and Roles

Super resolution - SRGAN

Anwesha Samaddar: Model training, fine tuning for SRGAN and performance measurement

Seungho Shin: Data preparation and preprocessing, preparing SRResNet

Object Detection - YOLO OBB

Manpreet Singh: YOLOv11-OBB label preparation, model training and performance measurement.

Shizra Tariq: Data preparation and evaluation for training YOLOv11

Thank you!!