Small-Object Detection in Remote Sensing Images and Video

Stamatios Orfanos

University of Piraeus NCSR Demokritos

October 29, 2024

Table of Contents

Introduction

Remote sensing imaging is a process used to gather information about objects or areas from a distance, typically using aircraft or satellites. Remote sensing imaging has applications across a broad spectrum of fields.

- Environmental monitoring
- Agriculture monitoring
- Disaster management
- Urban planning
- Military and intelligence

Urban Planning

Introduction

In remote sensing images the objects are small fraction of the pixels of the image, qualifying this process as Small Object Detection.

Compared with large and medium objects, small objects are more difficult to detect accurately for the following reasons:

- Small objects have low resolution and insufficient features
- The span of object-scale is large and multiple scales coexist
- The examples of small objects are scarce
- Categories for small objects are imbalanced for the majority of datasets

Introduction

There are two ways to define small objects in the context of object detection.

- Relative size, where the bounding box of a small object should cover less than 1% of the original image
- Absolute size, where a small object has size less than 32x32 pixels defined in MS-COCO dataset or 16x16 pixels defined in USC-GRAD-STDdb

Data and Data Preprocessing

The selected datasets cover a wide range of applications, from real-life scenarios to military and intelligence uses, ensuring a comprehensive evaluation of the detection models.

- Microsoft Common Object in COntext dataset
- Vis-Drone dataset
- Unmaned Arieal Vehicles - Small Object detection dataset

COCO2017 Dataset

The COCO2017 dataset includes complex everyday scenes with common objects in their natural context. It features:

- 80 object categories
- 118.000 training images
- 5.000 validation images
- 41.000 test images
- 1.5 million object instances
- Bounding boxes format: $[x_{center}, y_{center}, height, width]$
- Masks for objects provided
- Annotation format: Text format

Figure: VisDrone sample

COCO2017 Dataset

This dataset is used for object detection, segmentation, and captioning tasks. The class distribution of the dataset can be seen below:

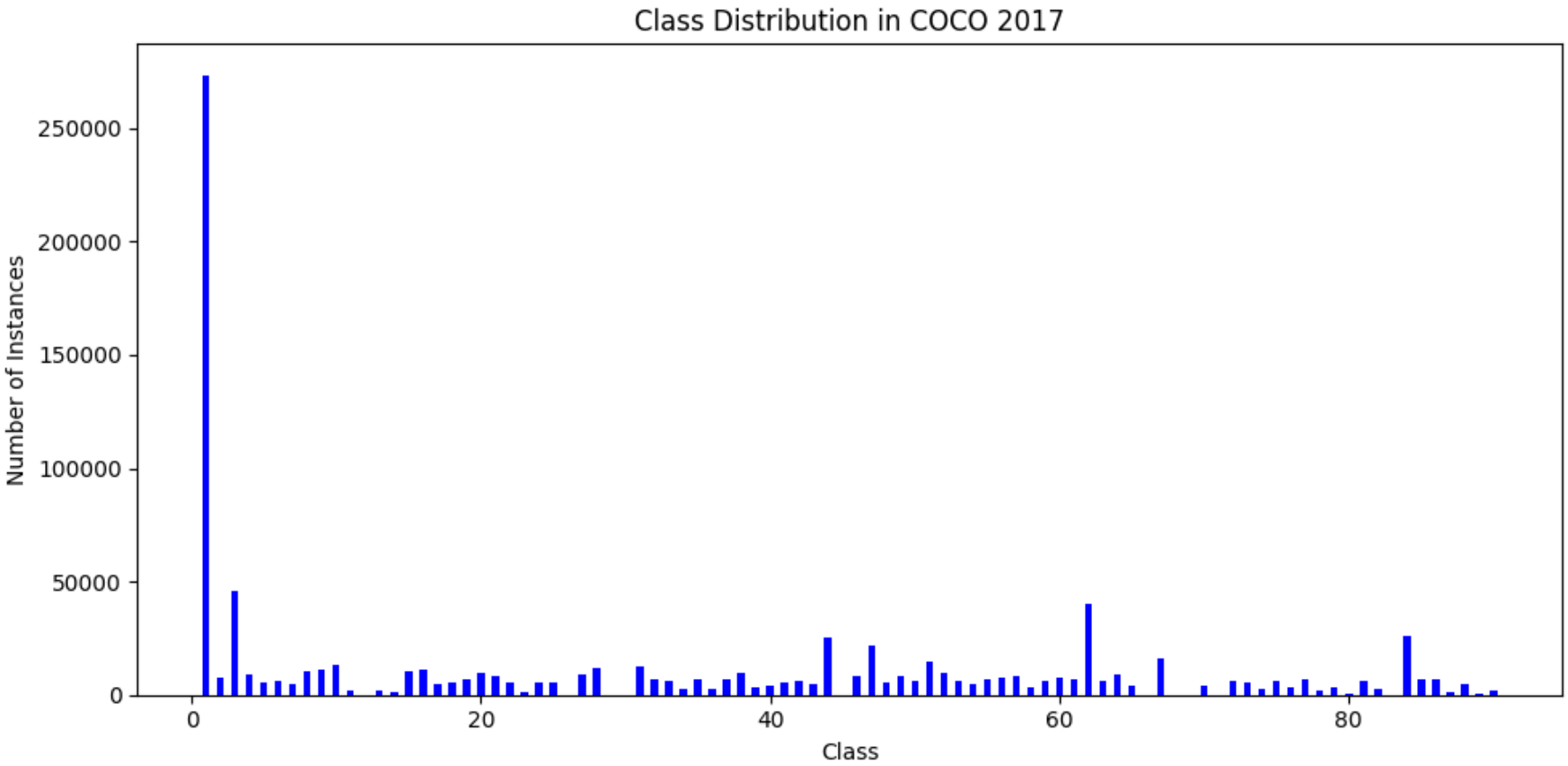

Figure: Class Distribution in COCO 2017

Vis-Drone Dataset

Vis-Drone is designed for drone-based image analysis and includes:

- 10 object categories
- 6.471 training images,
- 1.610 validation images
- 2.6 million object instances
- Bounding boxes format: $[x_{center}, y_{center}, height, width]$
- Masks for objects not provided
- Annotation format: Text format

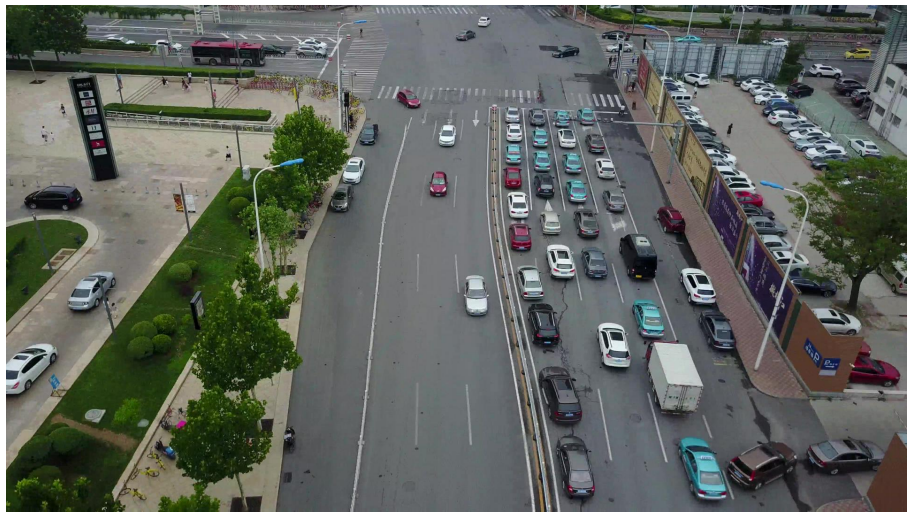

Figure: VisDrone sample

Vis-Drone Dataset

This dataset is used mainly for small object detection and segmentation tasks. The class distribution of the dataset can be seen below:

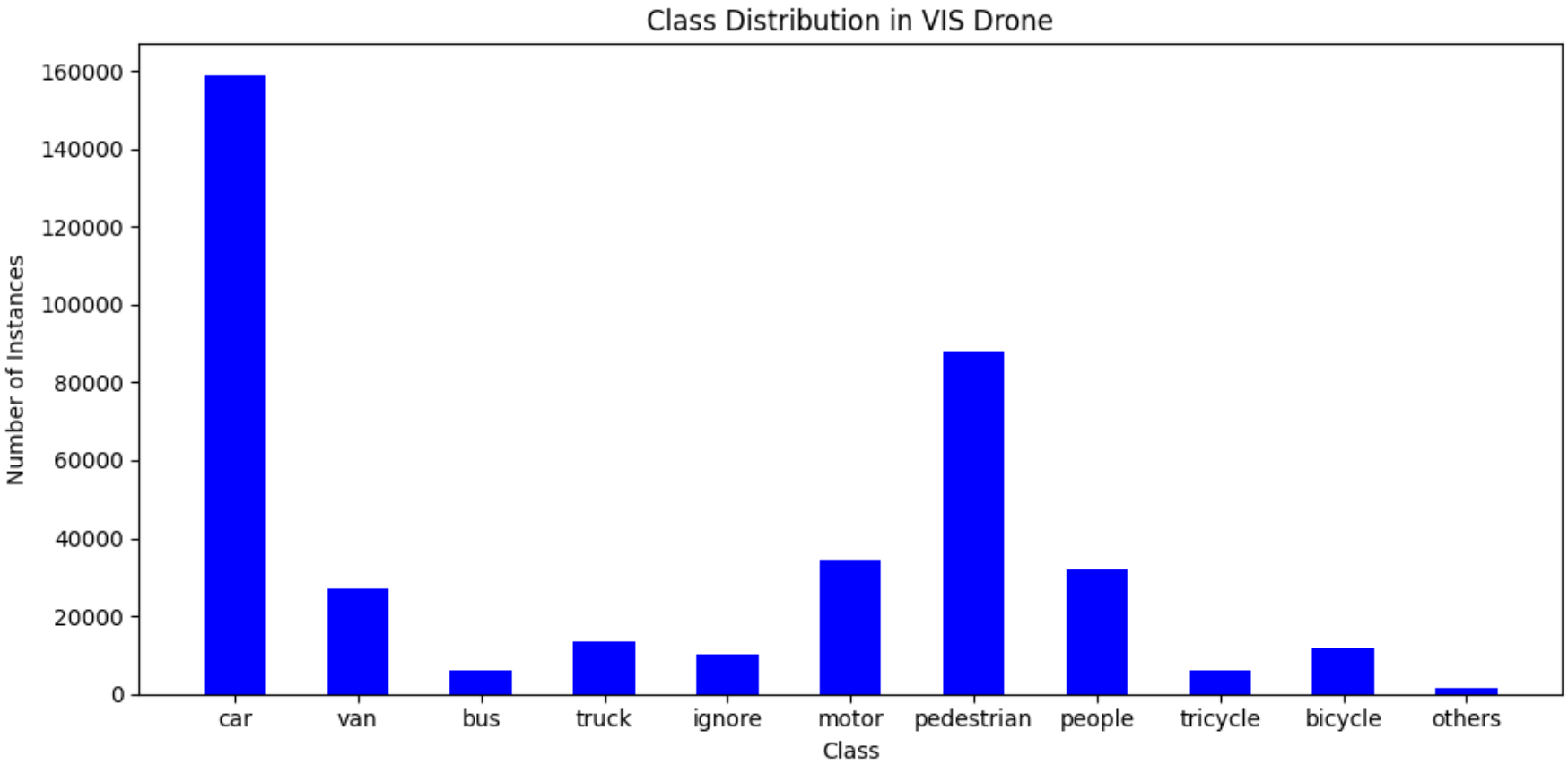

Figure: Class Distribution in Vis-Drone

UAV-SOD Dataset

The UAV-SOD dataset is targeted at small object detection from aerial perspectives, featuring:

- 10 object categories
- 717 training images
- 84 validation images
- 43 test images
- 18.234 object instances
- Bounding boxes format: $[x_{min}, y_{min}, x_{max}, y_{max}]$
- Masks for objects not provided
- Annotation format: XML format

Figure: UAV-SOD sample

UAV-SOD Dataset

This dataset is used mainly for small object detection. The class distribution of the dataset can be seen below:

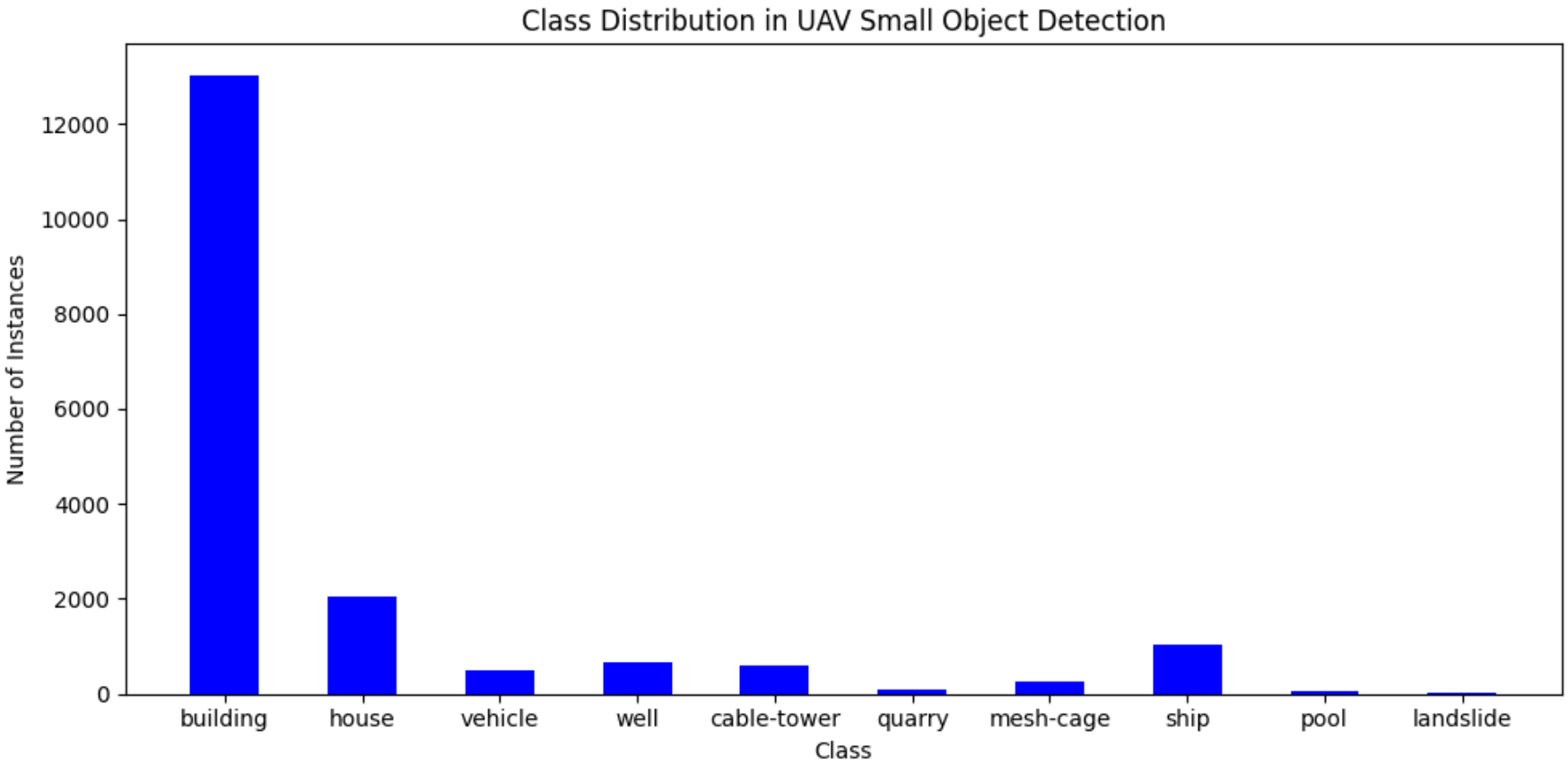

Figure: Class Distribution in UAV-SOD

Data Preprocessing Steps

Preprocessing is crucial for normalizing data and improving model training efficiency. Steps include:

- Resizing images and annotations to a uniform size of 600×600 pixels.
- Image padding to maintain aspect ratio without distortion.

Figure: Image before Preprocessing

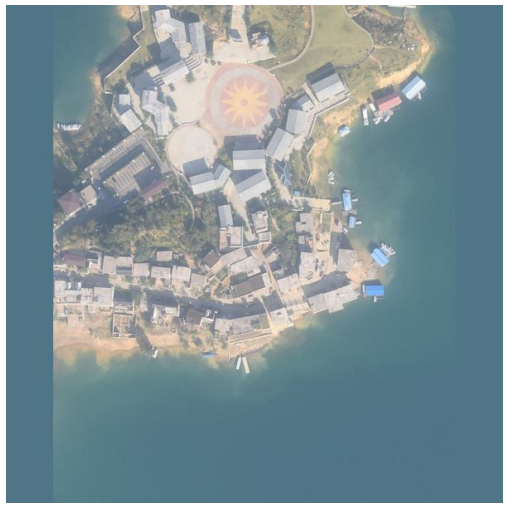

Figure: Image after Preprocessing

Data Preprocessing Steps

- Annotation format standardization for consistency across datasets.
- Normalization of image pixel values using dataset-specific mean and standard deviation.
- Create masks from bounding box coordinates.

Annotation Format Example:

$$x_{min}, y_{min}, x_{max}, y_{max}, class_{id}, [(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)]$$

Object Detection Metrics: Precision and Recall

Precision - The proportion of true positive identifications made by the model out of all positive identifications it made.

$$Precision = \frac{TP}{TP + FP}$$

Recall - The proportion of true positive identifications made by the model out of all actual positives available during the test.

$$Recall = \frac{TP}{TP + FN}$$

Object Detection Metrics: AP and mAP

Average Precision (AP) - A measure of precision across varying thresholds, reflecting the area under the precision-recall curve.

$$AP = \sum_{n} (R_n - R_{n-1}) P_n$$

Mean Average Precision (mAP) - The average of AP scores across all classes or varying IoU thresholds, providing a single overall effectiveness score for the detection system.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

Proposed Method

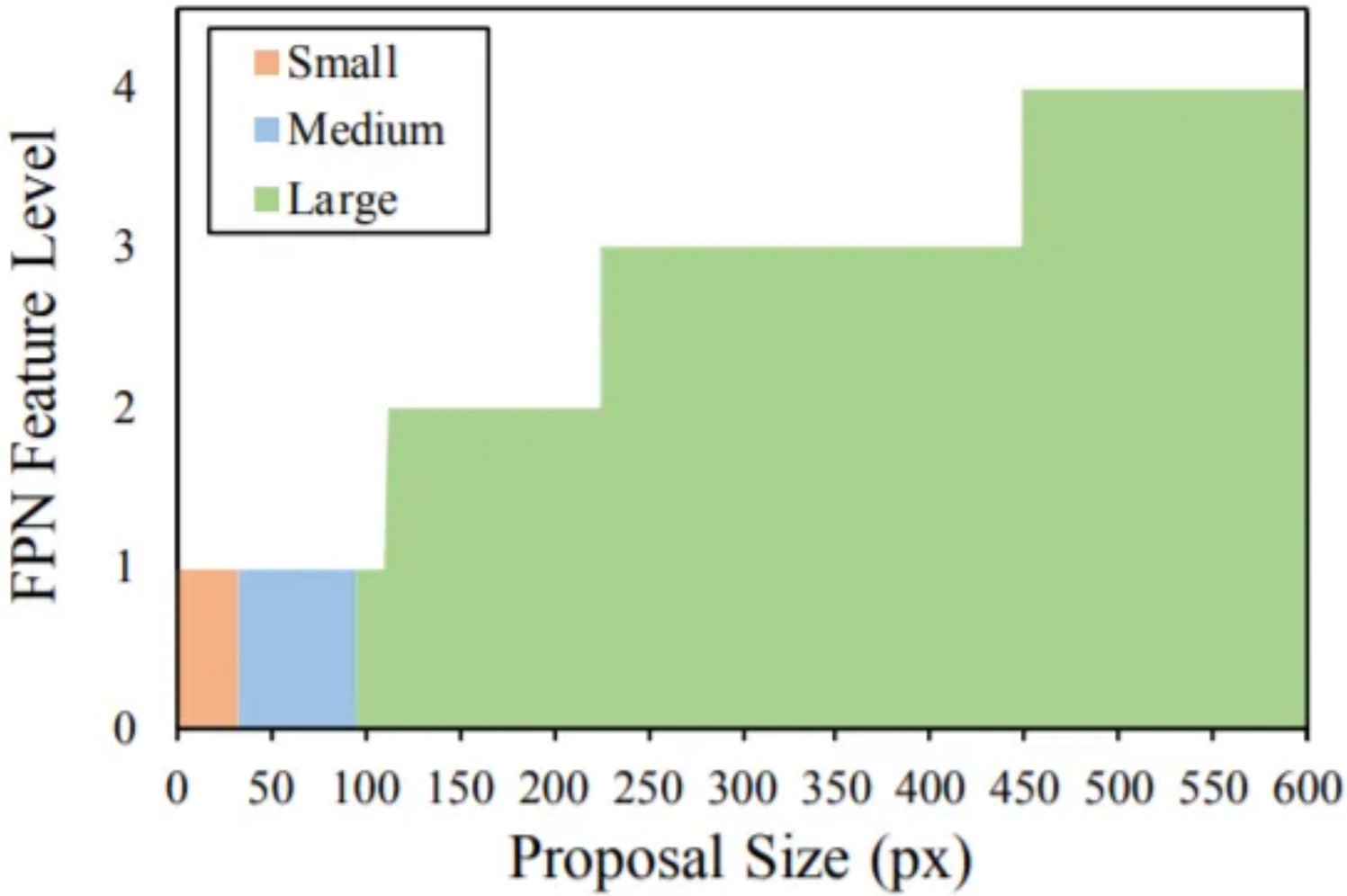

Figure: Object Detection Mapping

Proposed Method

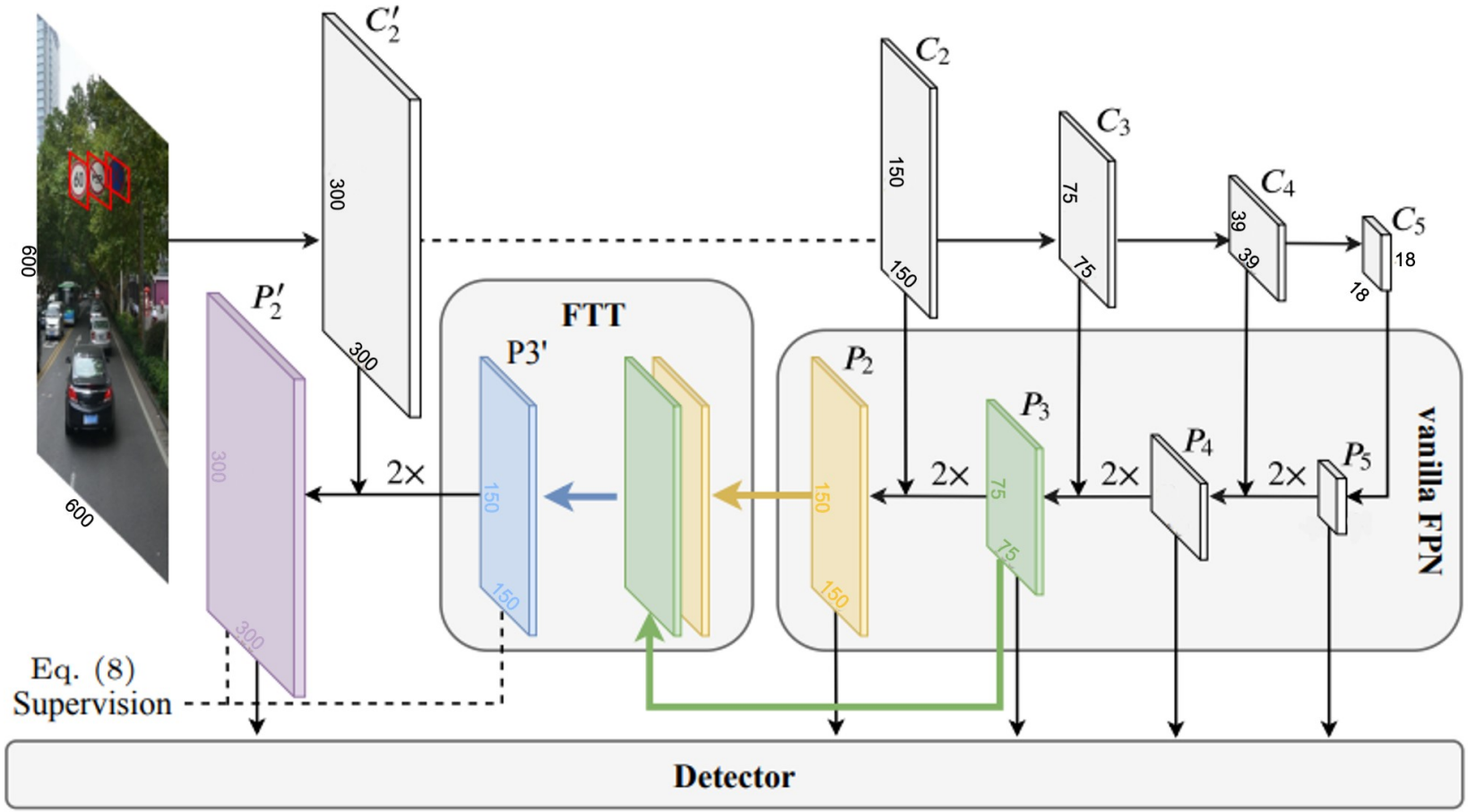

Figure: Extended Feature Pyramid Network

Proposed Method

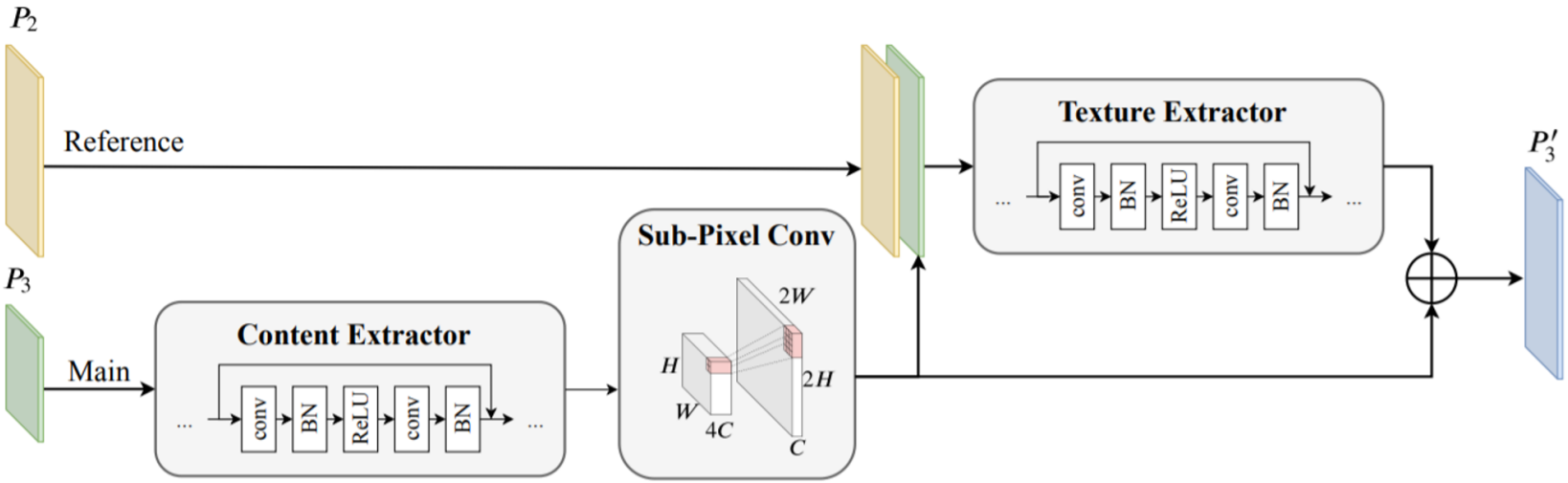

Figure: Feature Texture Transfer

Proposed Method

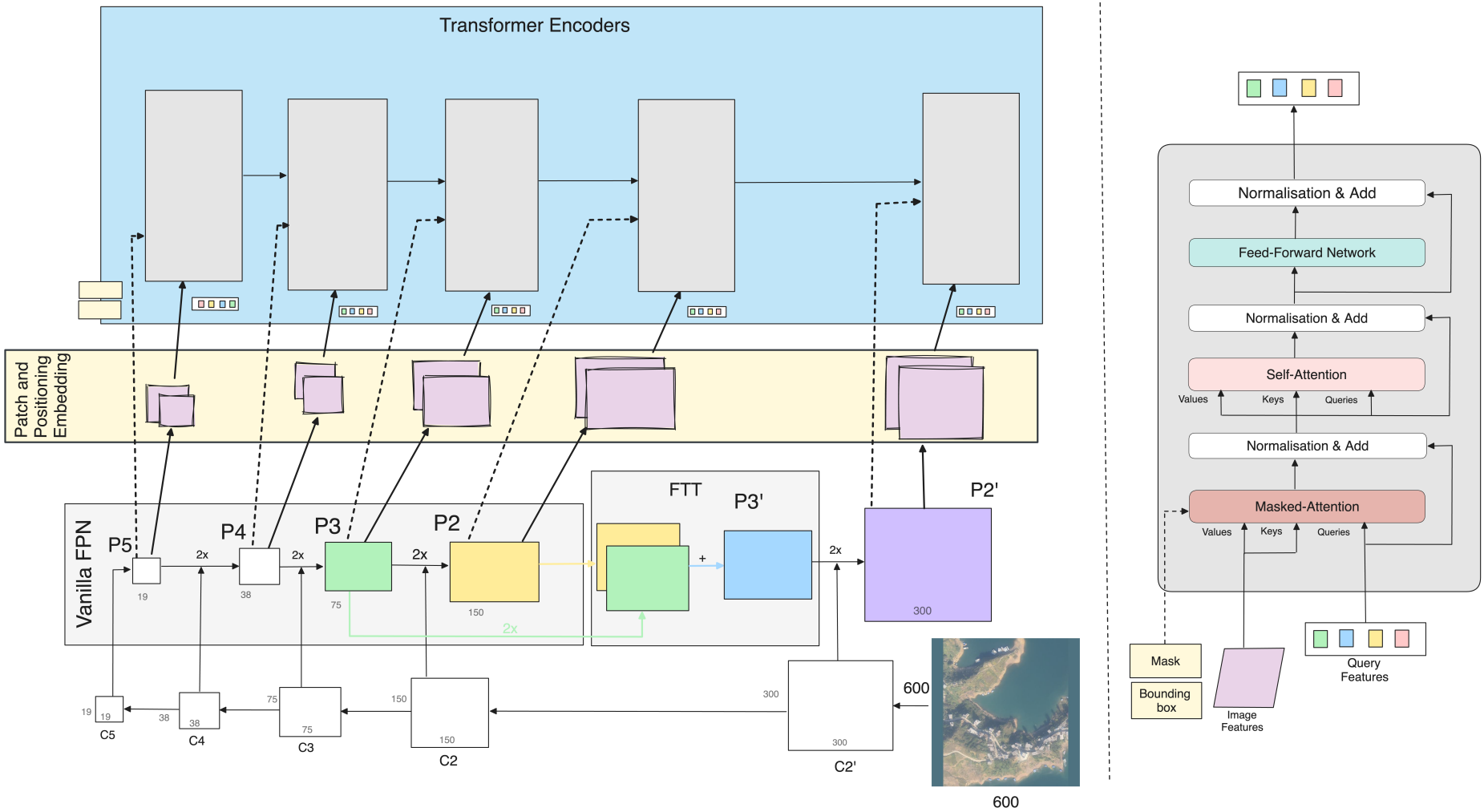

Figure: Extended Masked-Attention Mask Transformer Architecture

Proposed Method

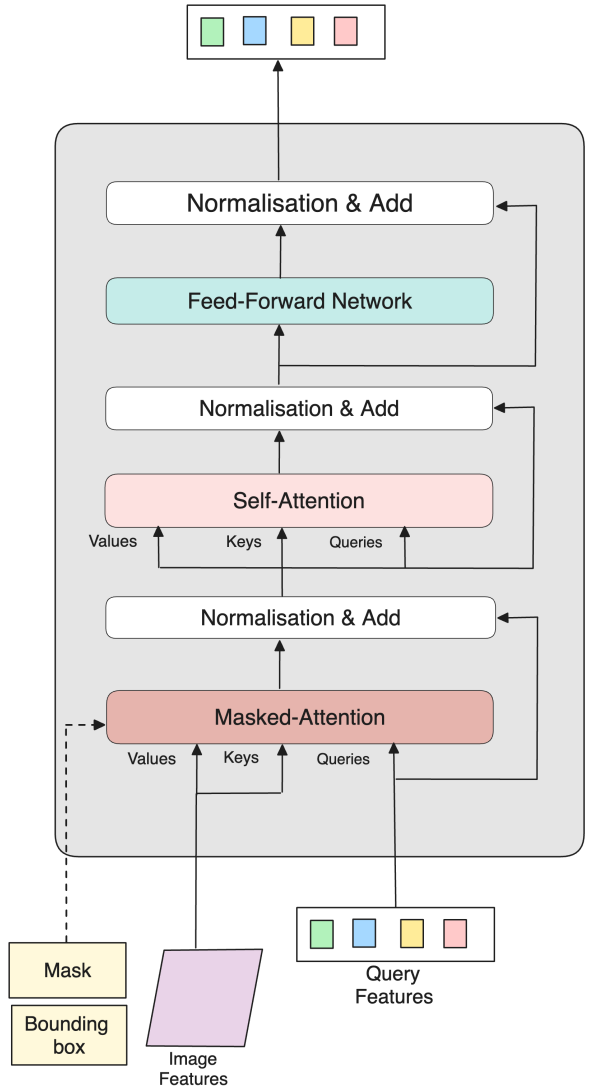

Figure: Vision Transformer Architecture

Proposed Method

To jointly optimize object detection and instance segmentation by incorporating multiple loss components, each addressing specific aspects of model performance.

$$L_{total} = \lambda_{mask} L_{mask} + \lambda_{bbox} L_{bbox} + \lambda_{class} L_{class}$$

- Mask Loss - $\lambda_{mask} = 1$
- Bounding Box Loss - $\lambda_{bbox} = 1$
- Class Loss - $\lambda_{class} = 0.8$

Experiments

To effectively implement and evaluate the Extended Masked-Attention Mask Transformer, we utilized Amazon Web Services (AWS) SageMaker, leveraging the powerful *ml.p4d.24xlarge* instance with eight NVIDIA A100 GPUs.

The table below summarizes the tailored training parameters per dataset, optimized through extensive experimentation to maximize model performance.

	MS COCO	**UAV-SOD**	**VisDrone**
Number of Epochs	125	75	75
Optimizer	AdamW	AdamW	AdamW
Learning Rate	1×10^{-4}	1×10^{-3}	1×10^{-3}
Batch size	4	4	4
Image size	600×600	600×600	600×600
Number of Anchors	30000	722	722

Table: Details of training parameters per dataset

Experiments - COCO2017

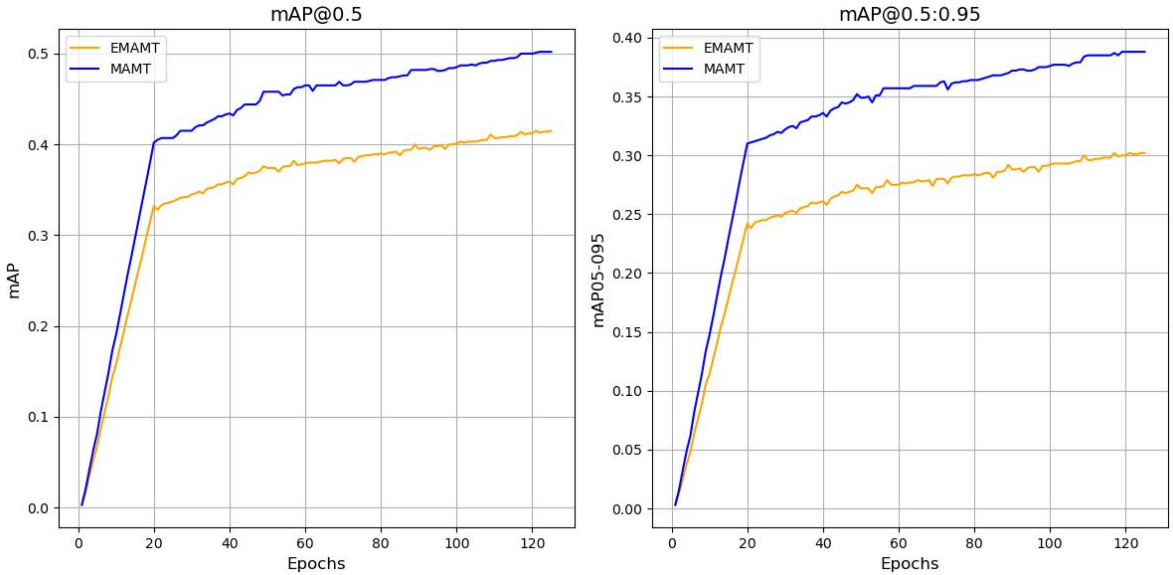

Figure: Performance comparison of MAMT and EMAMT on mAP at 50% IoU threshold (left) and the average mAP across IoU thresholds from 50% to 95% (right) for the COCO2017 dataset

Model	mAP(@0.5)	mAP(@0.5:0.95)	Queries	Parameters	GFLOPs
EMAMT	44.5	30.2	**100**	**95M**	**492**
MAMT	**50.2**	**38.8**	200	216M	868

Table: Results for COCO dataset

Experiments - COCO2017

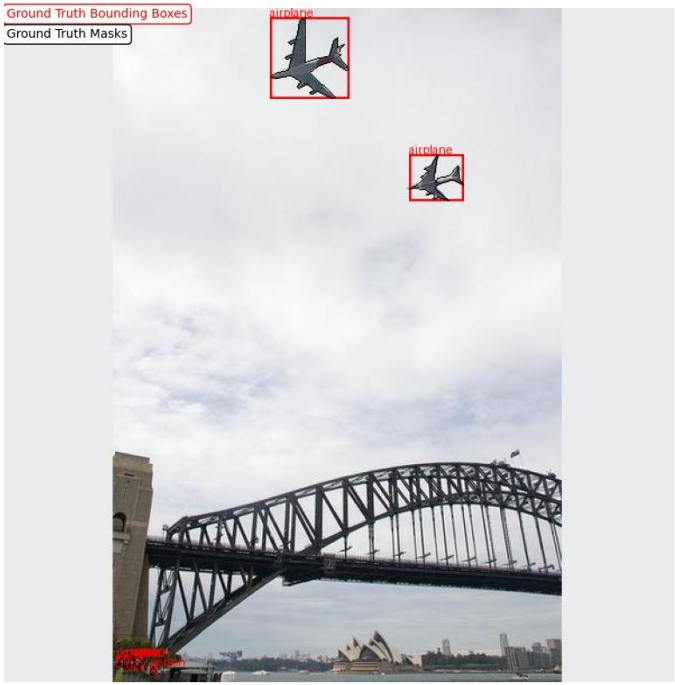

Figure: Image with Ground Truth Bounding Boxes and Masks

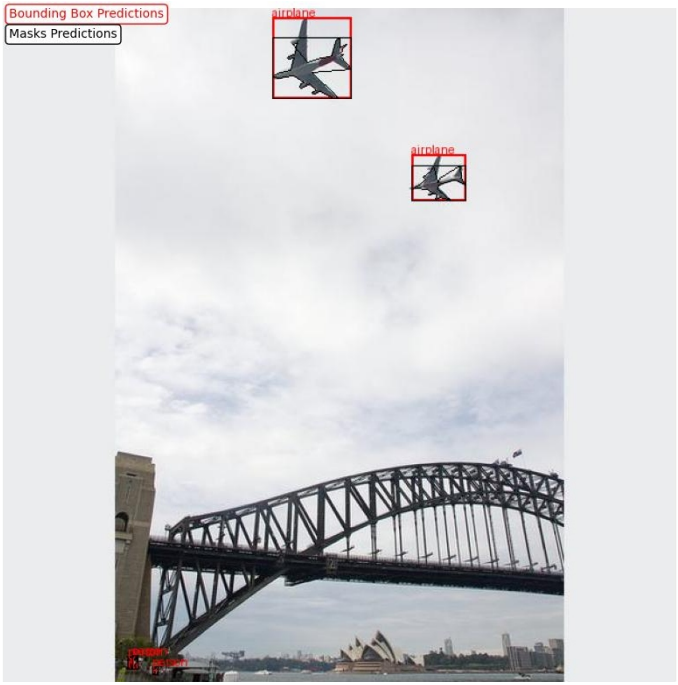

Figure: Image with Predicted Bounding Boxes and Masks

Experiments - VisDrone

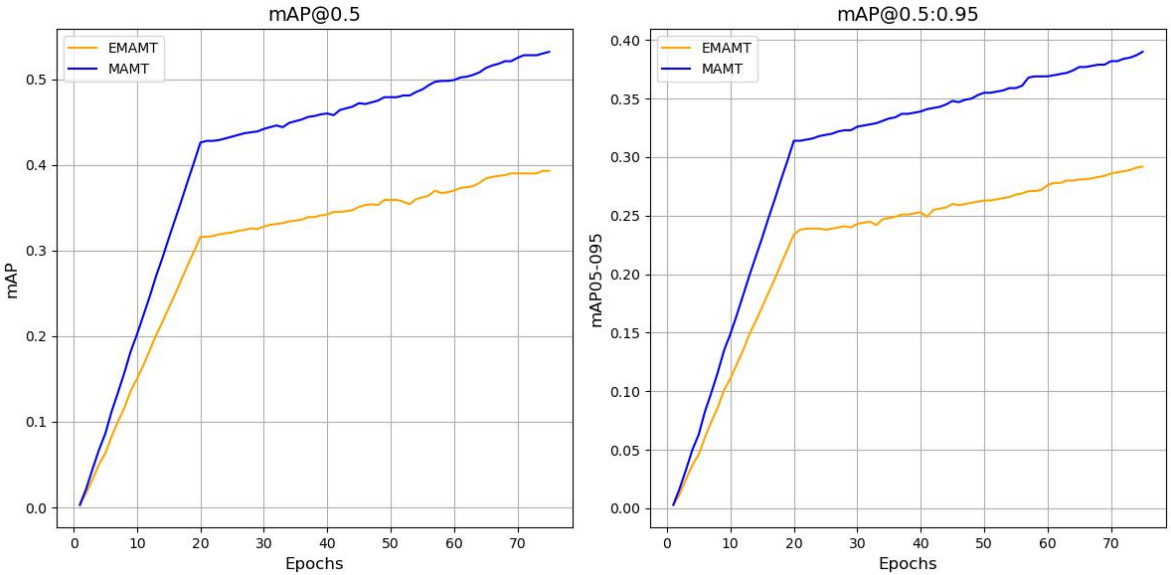

Figure: Performance comparison of MAMT and EMAMT on mAP at 50% IoU threshold (left) and the average mAP across IoU thresholds from 50% to 95% (right) for the VisDrone dataset

Model	mAP(@0.5)	mAP(@0.5:0.95)	Queries	Parameters	GFLOPs
EMAMT	39.5	29.2	**100**	**95M**	**492**
MAMT	**53.2**	**39.2**	200	216M	868

Table: Results for VisDrone dataset

Experiments - VisDrone

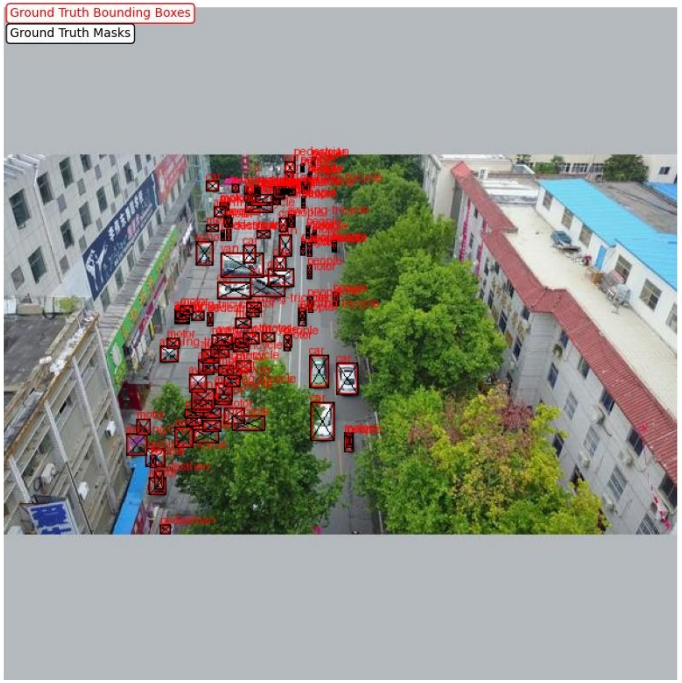

Figure: Image with Ground Truth Bounding Boxes and Masks

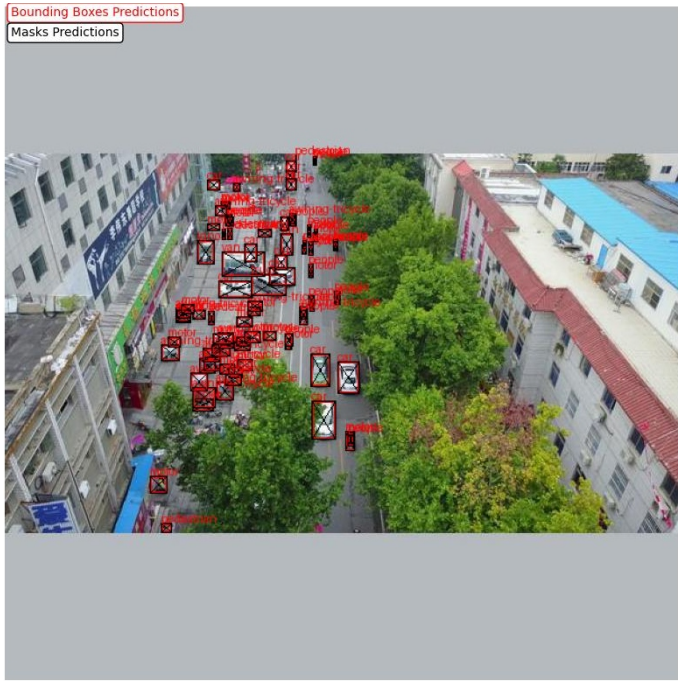

Figure: Image with Predicted Bounding Boxes and Masks

Experiments - UAV-SOD

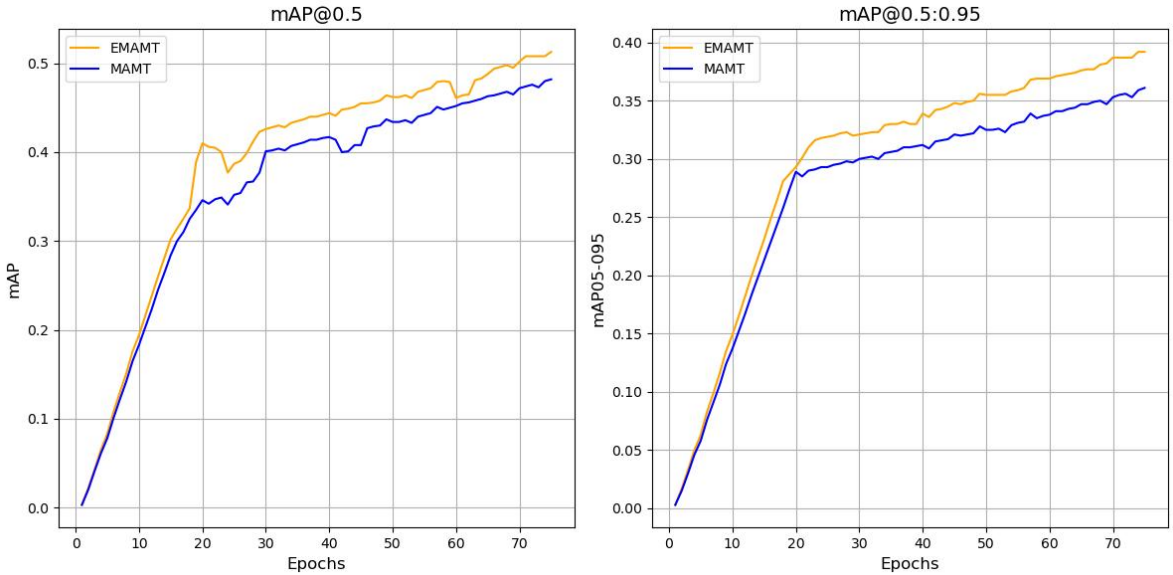

Figure: Performance comparison of MAMT and EMAMT on mAP at 50% IoU threshold (left) and the average mAP across IoU thresholds from 50% to 95% (right) for the UAV-SOD dataset

Model	mAP(@0.5)	mAP(@0.5:0.95)	Queries	Parameters	GFLOPs
EMAMT	**51.3**	**39.2**	**100**	**95M**	**492**
MAMT	48.2	36.1	200	216M	868

Table: Results for UAV-SOD dataset

Experiments UAV-SOD

Figure: Image with Ground Truth Bounding Boxes and Masks

Figure: Image with Predicted Bounding Boxes and Masks

Conclusion: Overview and Achievements

We introduced the *Extended Masked-Attention Mask Transformer (EMAMT)*, integrating the Enhanced Feature Pyramid Network (EFPN) with Mask2Former, aiming at enhanced efficiency and accuracy in object detection.

Key Findings:

- EMAMT achieved up to 56% reduction in model complexity compared to the traditional Masked-Attention Mask Transformer (MAMT).
- EMAMT surpassed MAMT in performance on the UAV-SOD dataset, affirming its effectiveness in aerial small object detection.
- In more diverse environments like MS COCO and VisDrone, EMAMT showed reductions in mAP by 6.5% and 13.7%, highlighting areas for further optimization.

Future Work

Planned Investigations and Optimizations:

- Detailed analysis of each feature map in EMAMT to assess their individual contributions and optimize model structure.
- Explore strategies to further reduce computational demands while maintaining accuracy, crucial for real-time applications.

Thank you for your attention and time.