Small-Object Detection in Remote Sensing Images and Video

Stamatios Orfanos

University of Piraeus NCSR Demokritos

October 8, 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:

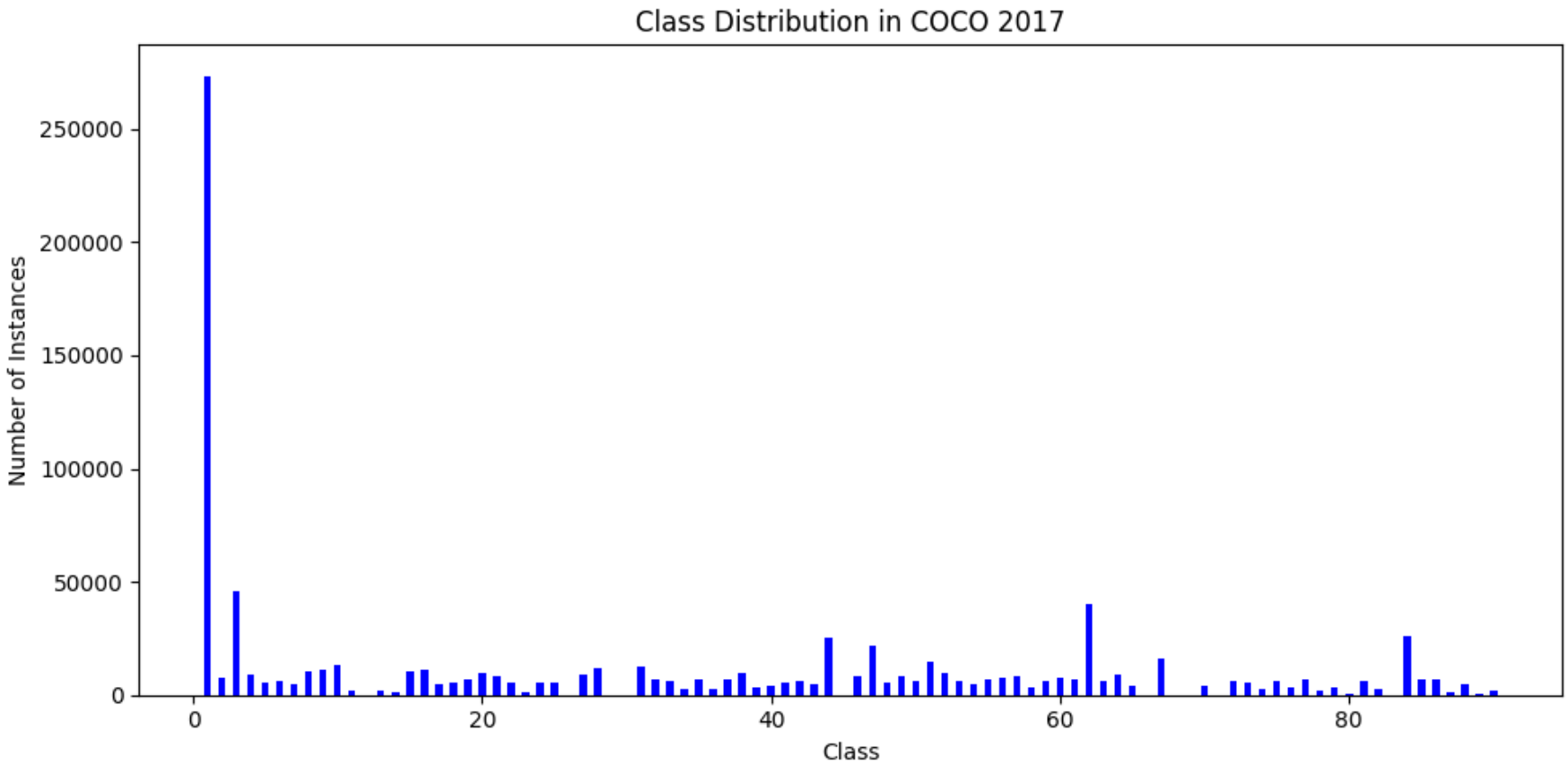

Figure: Class Distribution in COCO 2017

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

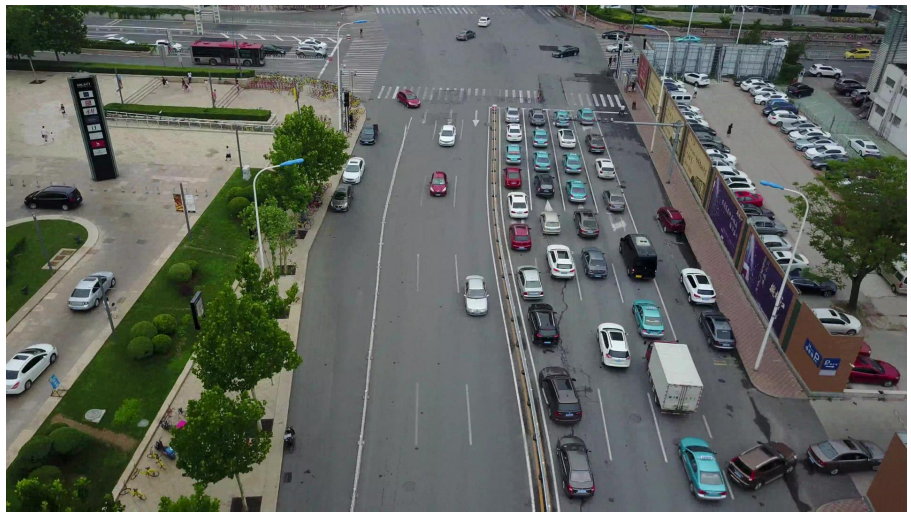

Figure: VisDrone sample

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:

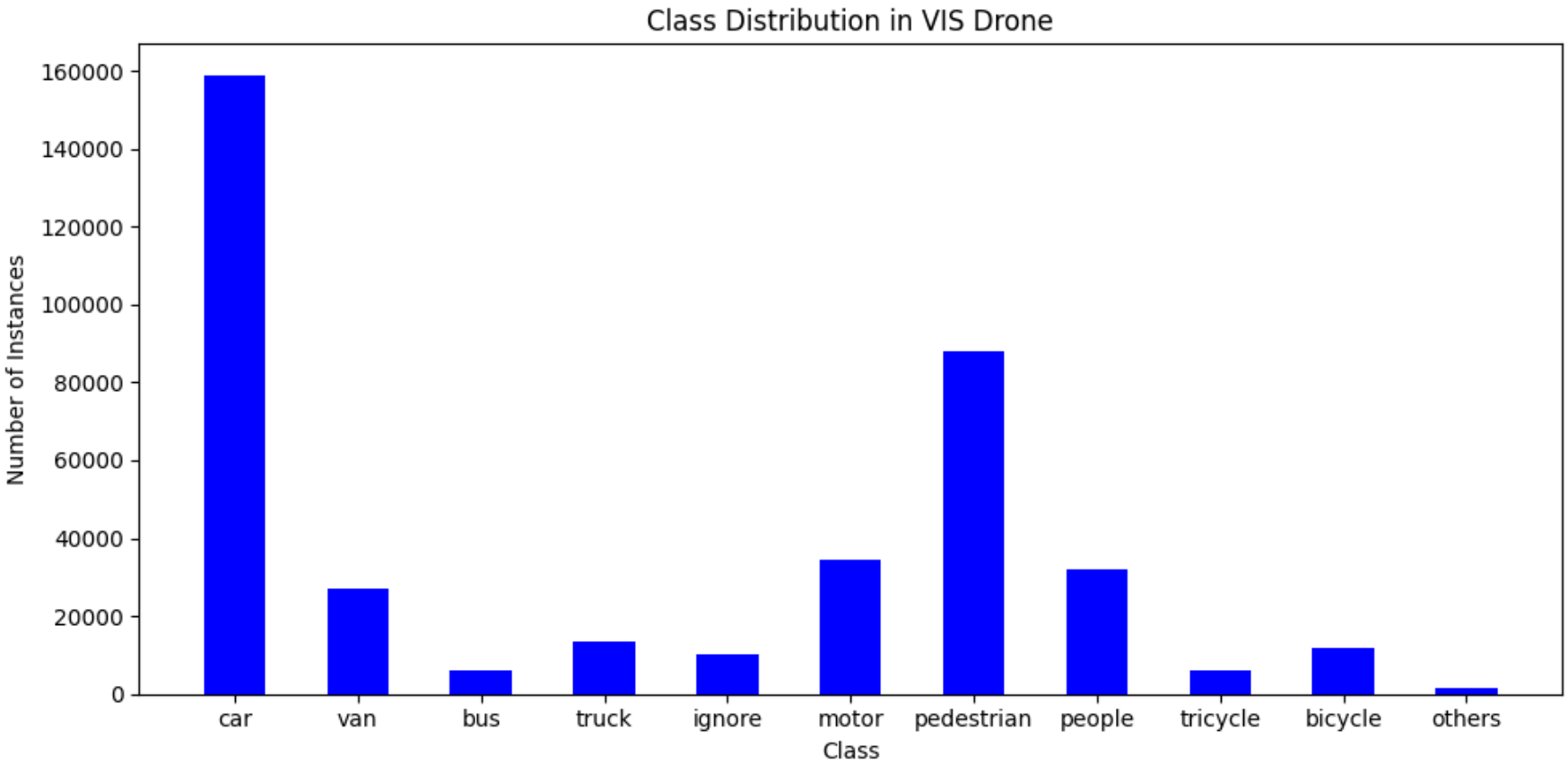

Figure: Class Distribution in Vis-Drone

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:

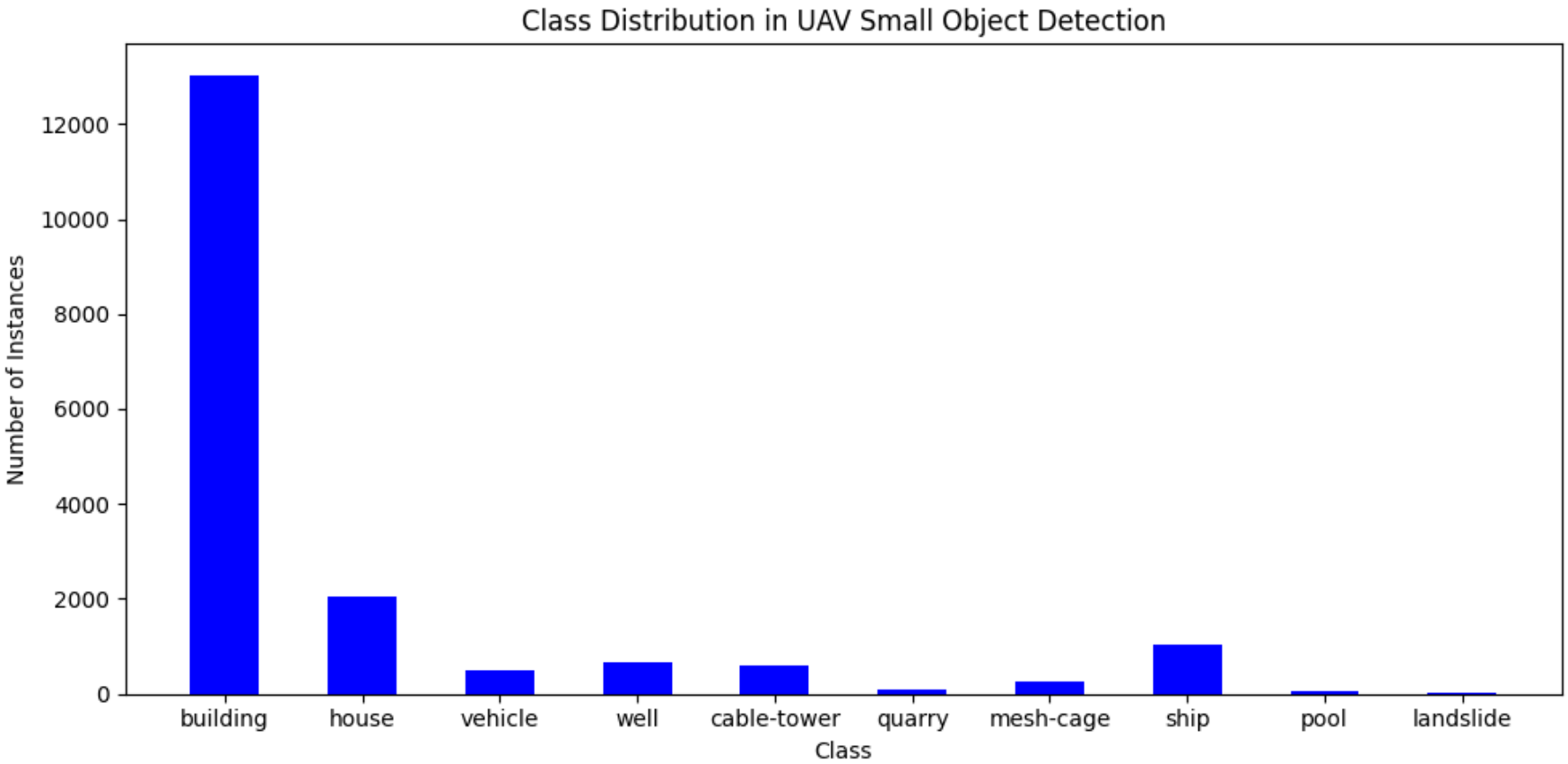

Figure: Class Distribution in UAV-SOD

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

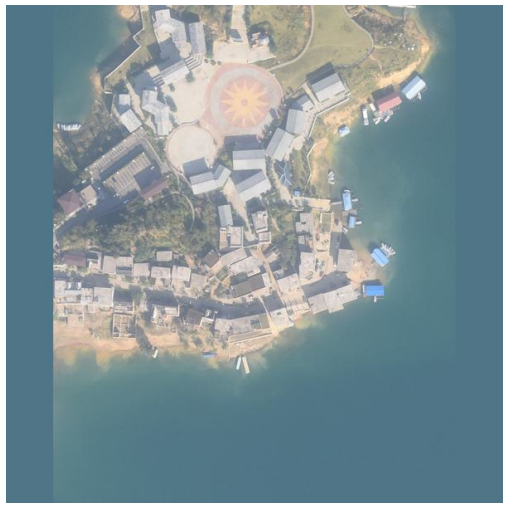

Figure: Image after Preprocessing

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

Object Detection Metrics

Proposed Method

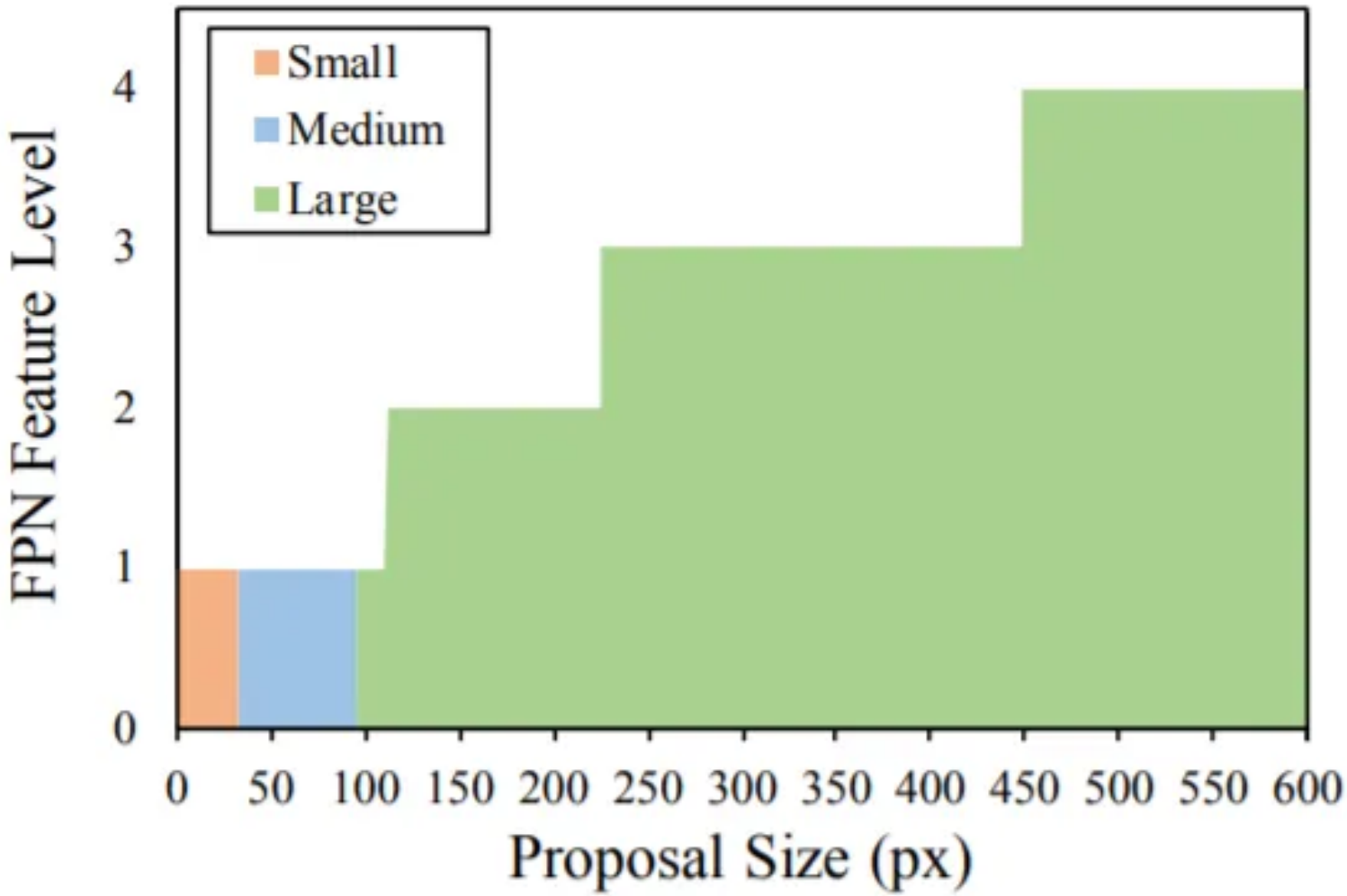

Figure: Object Detection Mapping

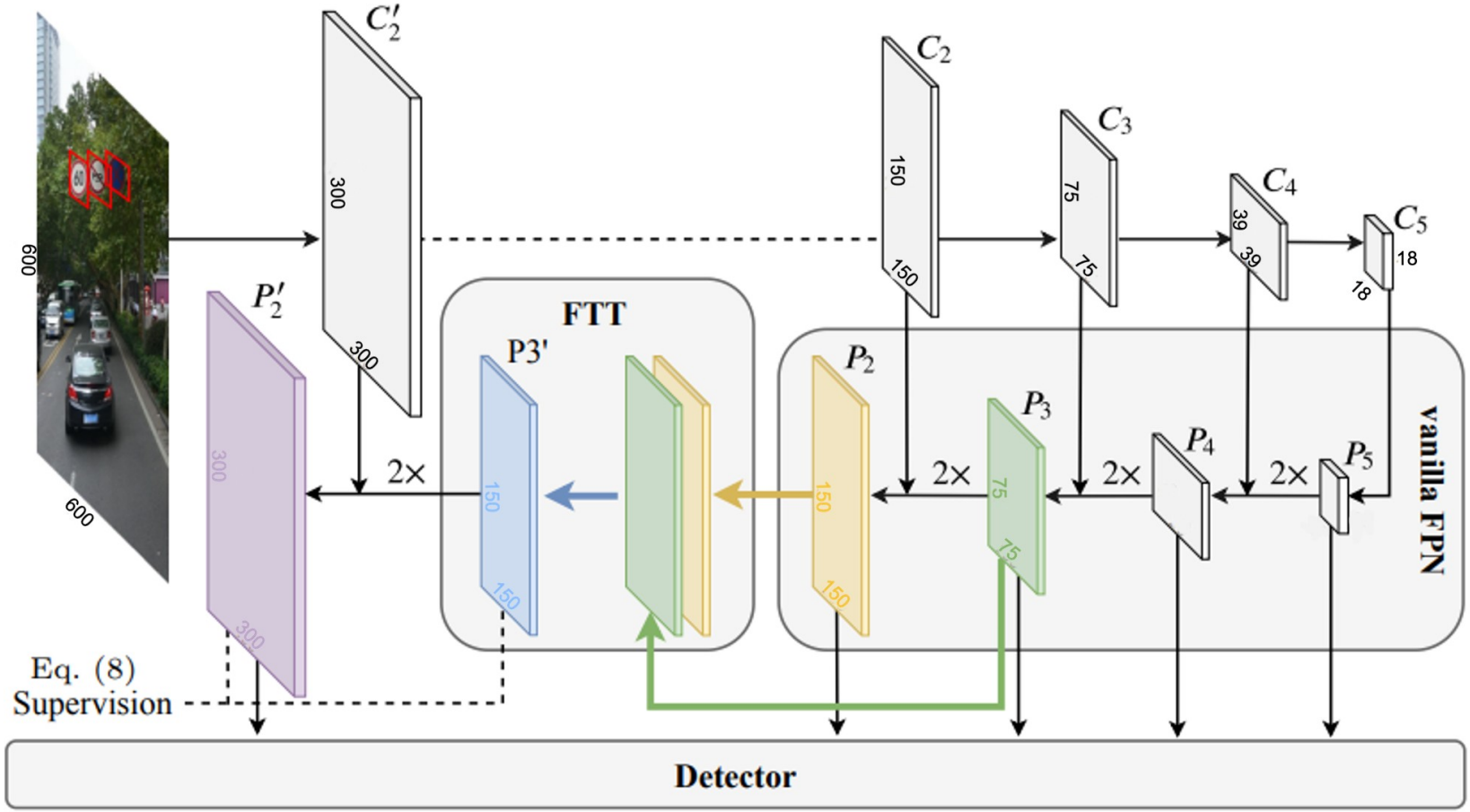

Figure: Extended Feature Pyramid Network

Proposed Method

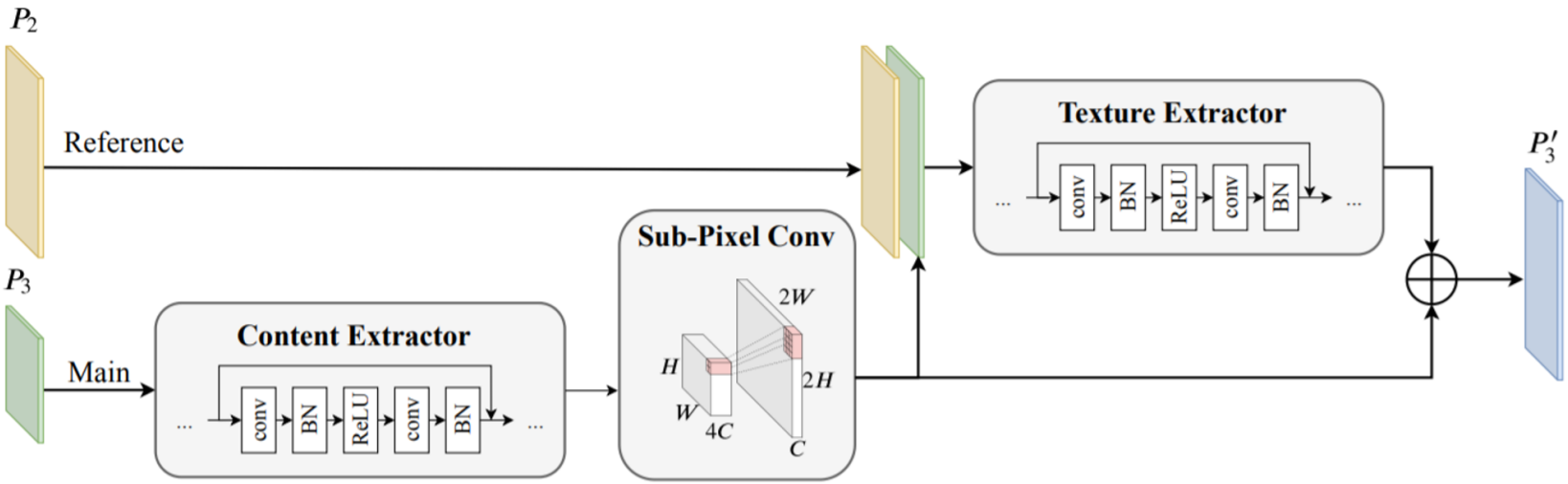

Figure: Feature Texture Transfer

Proposed Method

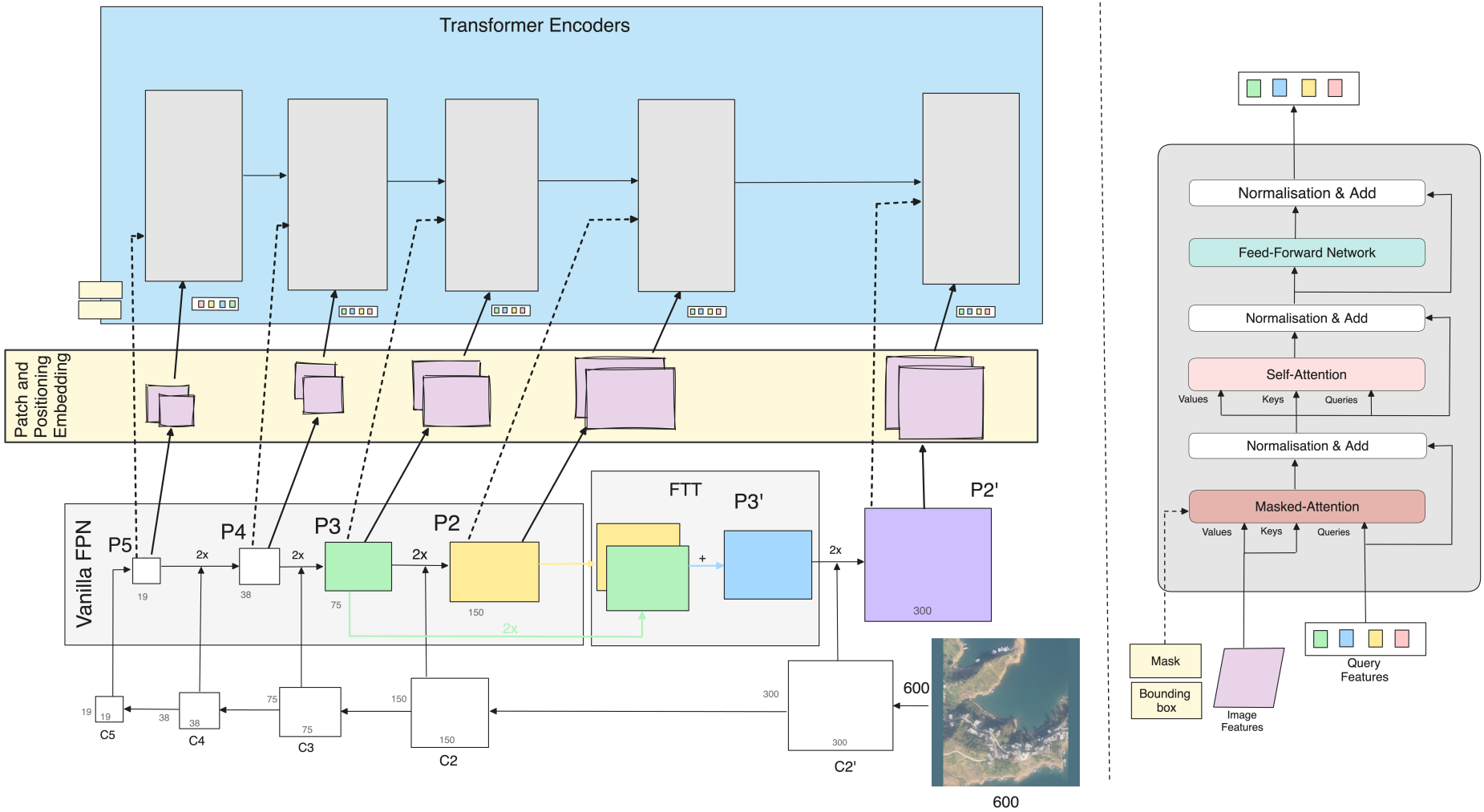

Figure: Extended Masked-Attention Mask Transformer Architecture

Proposed Method

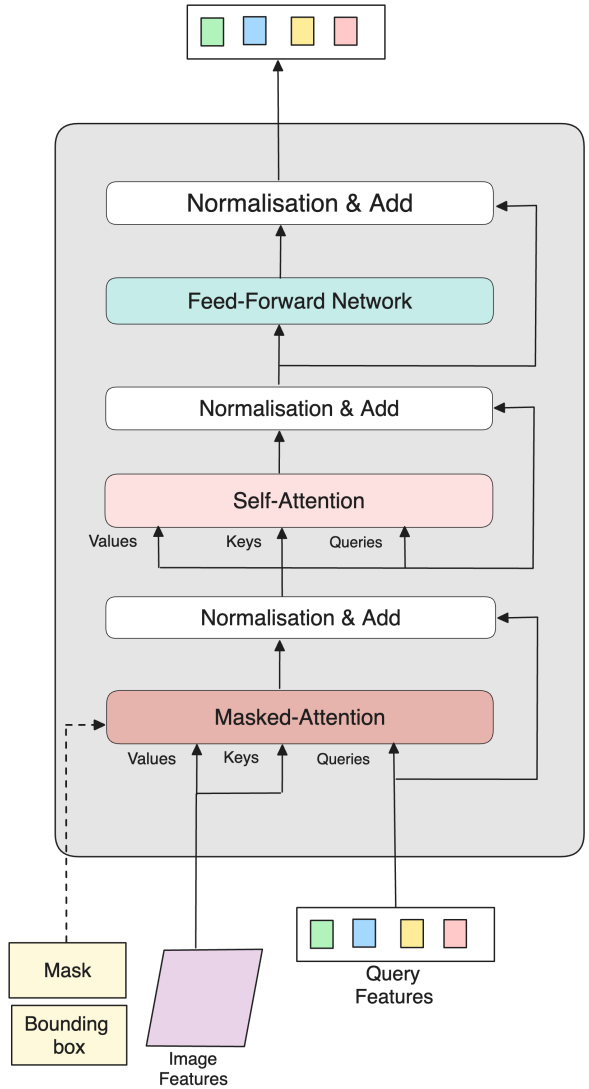

Figure: Vision Transformer Architecture

Experiments

Conclusion