

# Tree Detection using YOLO

## Overview

A system for detecting trees in aerial/drone images using YOLO model (YOLO v11) at multiple resolutions. This approach enhances detection across varying object sizes by processing images at different scales. Features include:

- Multi-scale detection: process images at 5 resolutions (1.0x, 0.75x, 0.50x, 0.30x, 0.25x)
- Tiled processing: Handle large images through overlapping tiles
- Outputs: CSV, GeoJSON, and visualization files (.jpg)
- Coordinate transformation: Convert between pixel, image, and geographic coordinates

## Usage

- **Command line Interface**

```
python detection.py --image_path /path/to/image.tif --model_path /path/to/model.pt --output_dir /path/to/output --conf_threshold 0.25
```

Parameter	Type	Default	Description
--image_path	Str	Required	Path to input geoTIFF/image file
--model_path	Str	Required	Path to YOLO model weights (.pt file)
--output_dir	Str	Required	Directory to save the output files
--conf_threshold	float	0.25	Confidence threshold for detection

- **Core Functions**

**detect\_at\_multiple\_resolutions()** : main function that performs multi-resolution tree detection

**params:**

- **image\_path (str)** : path to input image
- **model (YOLO)** : loaded yolo model instance
- **output\_dir (str)** : output directory file
- **tile\_size (int)** : size of processing tile (default: 640)
- **overlap (int)** : size of overlap between tiles (default 32)
- **resolutions (list)** : scale factor for multi-resolution processing
- **conf\_threshold (float)** : detection confidence threshold

**returns:**

- **all\_points (list)** : point object in geographic coordinates
- **detection\_points\_pixel (list)** : detection points in pixel coordinates
- **detection\_boxes (list)** : bounding box coordinates
- **counts\_per\_tile (list)** : detection count per tile

## Algorithm Details

- Multi-resolution Strategy

resolutions = [1.0, 0.75, 0.5, 0.3, 0.25] # Scale factors

- Tiling Process

```
for y in range(0, new_height, tile_size - overlap):
    for x in range(0, new_width, tile_size - overlap):
```

- Tile Size: 640x640 pixels (standard YOLO input size)
- Overlap: 32 pixels to prevent edge artifacts
- Edge Handling: Skips tiles smaller than 50% of tile size

## Output Files

- Data Files

File	Format	Description
multi_resolution_detections.csv	CSV	Detection points with metadata
multi_resolution_points.geojson	GeoJSON	Geographic coordinates of detections
multi_resolution_boxes.csv	CSV	Bounding box coordinates
multi_resolution_tile_counts.csv	CSV	Detection statistics per tile

- Visualization Files

File	Description
multi_resolution_detection.png	Individual resolution detection maps
combined_multi_resolution.png	All detections overlaid on original image

## Terminal Summary

=== MULTI-RESOLUTION DETECTION SUMMARY ===

Resolution 1.0x: 156 detections, avg confidence: 0.834

Resolution 0.75x: 142 detections, avg confidence: 0.819

Resolution 0.5x: 128 detections, avg confidence: 0.802

Resolution 0.3x: 95 detections, avg confidence: 0.791

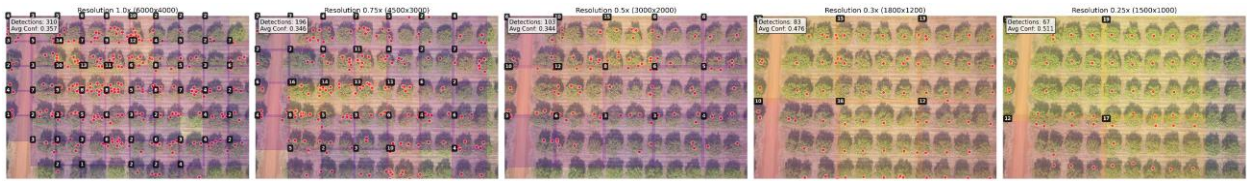
Resolution 0.25x: 87 detections, avg confidence: 0.785

Total unique detections: 608

Overall average confidence: 0.806

## Sample Output

multi\_resolution\_detection.png



combined\_multi\_resolution.png

