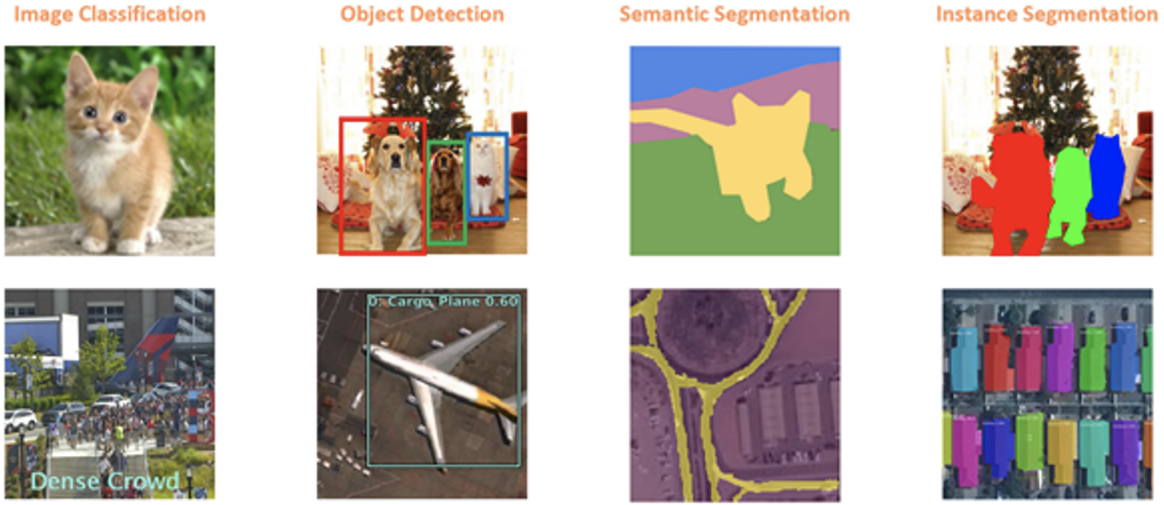
Object Detection of Small Vehicles in Satellite Imagery

David English, Thomas Keeley

Computer Vision & Satellite Imagery

- Familiar applications can be applied to satellite imagery
- Large volumes of available satellite at very-high resolutions available
- Several target applications
- Significant amount of data and cost of computing



Problem Statement

Can object detection frameworks be applied and evaluated in detecting small vehicles in very-high resolution satellite imagery?

Project Layout

Data Exploration → Framework Implementation → Comparative Results

Evaluate data

- Types of classes
- Annotation format
- Volume
- Image resolution

Implementation

- Benchmark object detection
- Advantages and assumptions
- Plan for this project

Results

- Accuracy
- Speed
- Ease of implementation
- Limitations

Data

DOTA: A Large-scale Dataset for Object Detection in Aerial Images

Source

- Images collected from Google Earth
- Annotated by experts in aerial image interpretation using 16 common object categories

Categories

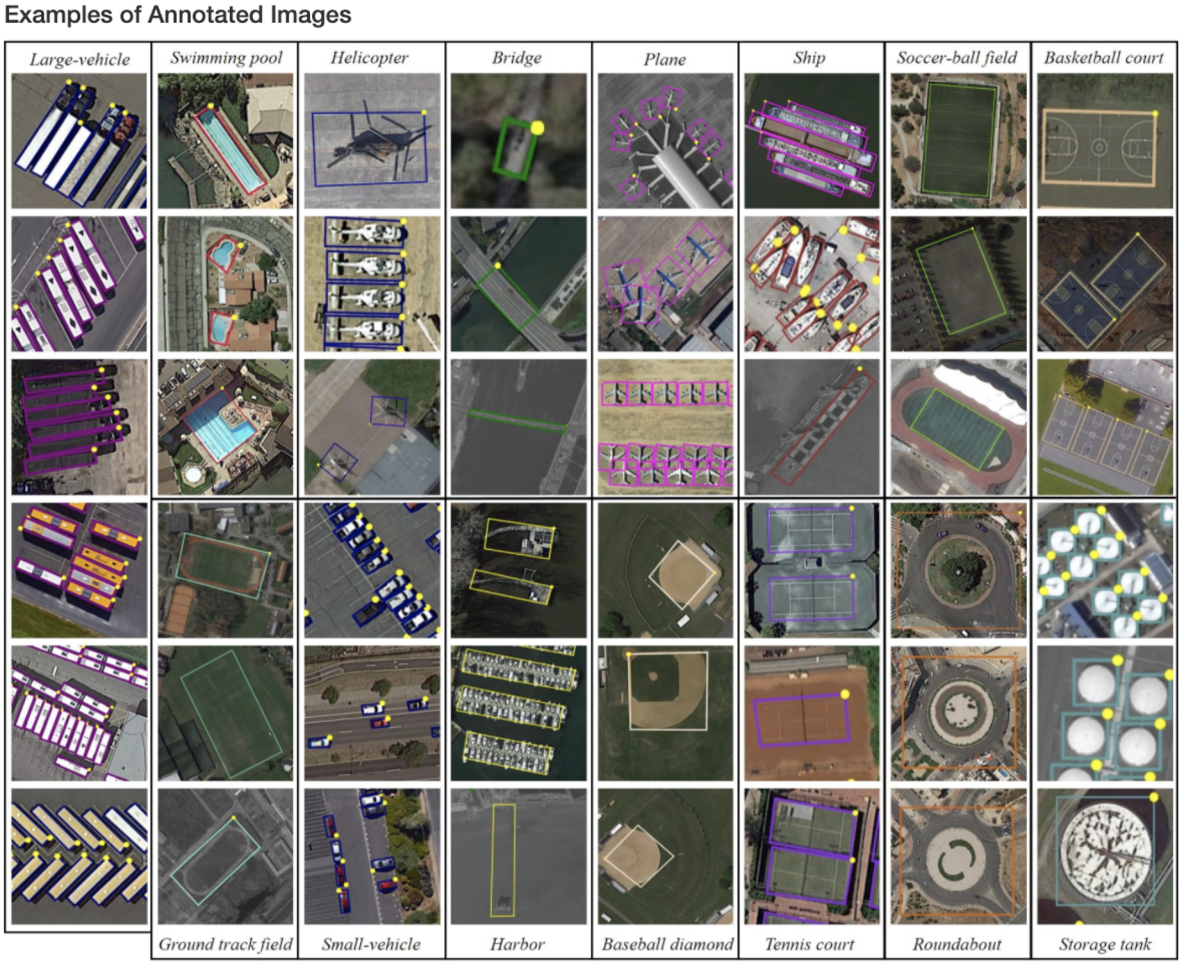
- plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool and container/crane

Annotations

- Oriented annotations of objects
- Capture size, shape and directionality

DOTA

- For this project, the focus will be placed solely on small vehicles.
- The oriented annotations are refactored to reflect horizontal bounding boxes



Implementation

Object Detection Frameworks and Implementation

Single Stage vs. Two Stage

- Single stage ingests target image through convolutional layers and make prediction
- Two stage first conducts regional proposals for targeting then conducts prediction

Implementations in this project

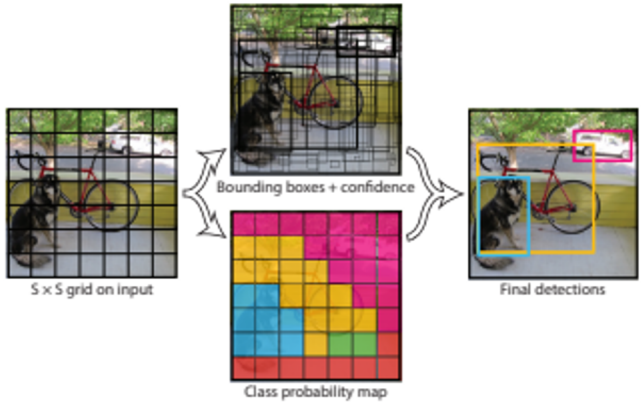
- YOLO
- Retinanet
  - Resnet18
  - Resnet50
  - Resnet101

Plan for comparison

- The balance between accuracy and speed
- Ease of implementation
- Mean average precision (mAP)

YOLO

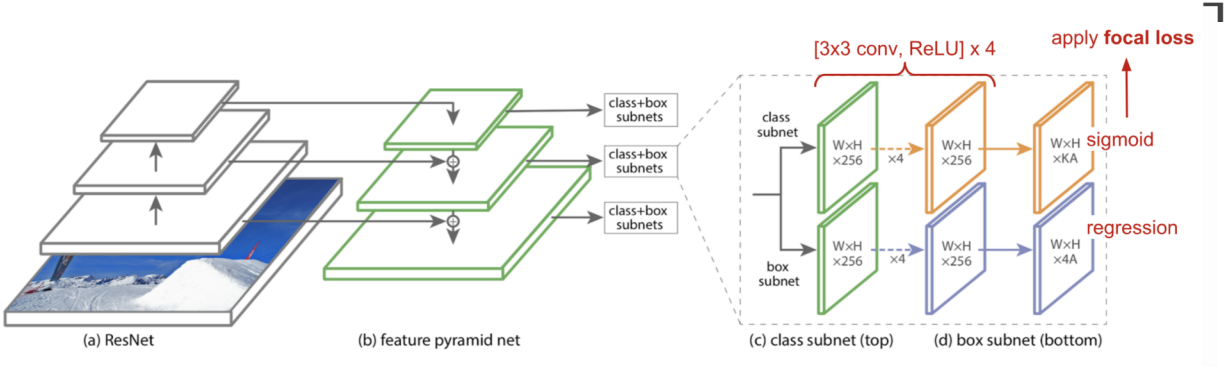
- Designed to not have separate detection and classification steps
  - Speed
  - Generalizable
  - Reduced Background Errors
- Divide image into grid boxes
  - For each grid box predict:
    i. Odds that a object is centered in the box and confidence in that box's boundaries
    ii. A classification of the object in the box independent of the likelihood there is an object in the box
    iii. Struggles when multiple objects per box



- Standard Yolo consists of 24 Convolutional layers followed by 2 fully connected layers
  - Fast-YOLO reduces that to only 9 Convolutional Layers

Retinanet

- Single Stage Detector
- Built with ResNet backbone
- Dense target object detection



- Feature pyramid network
  - Collection of convolutional layers that aim to capture features at different scales and sizes
- Focal Loss
  - Addresses the class imbalance between target objects and image background
    - Greater focus placed on objects by increasing weights for hard to classify pixels. Background weights are discounted. Increased accuracy without compromising speed.

Results

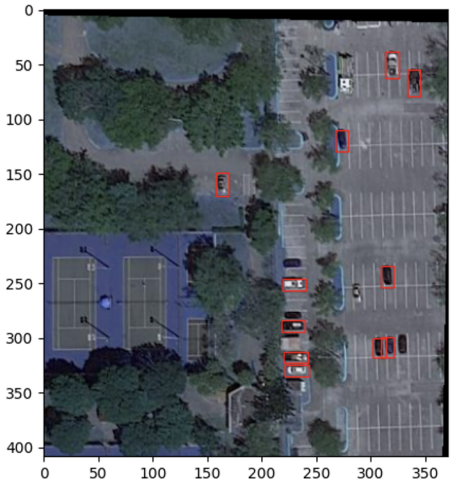
Comparative Metrics

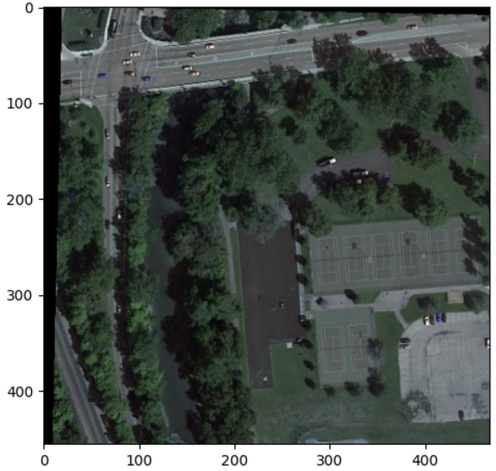
- Clear tradeoff in speed and accuracy with Retinanet/Resnet backbone
- With added depth and model complexity comes increased accuracy
- Fast-YOLO's training time was substantially longer than Retinanet, but inference speed was much shorter
- Fast-YOLO's accuracy is much lower than Retinanet
    - Object Density

Model	Training Time	Inference Time	mAP
Retinanet - Resnet18	00:03:15:00	127.2	0.44
Retinanet - Resnet50	00:05:30:00	178.9	0.47
Retinanet - Resnet101	00:06:45:00	200.9	0.55
Fast YOLO	24:00:00	26.3	.0046

Visual Results

- Displayed predictions shows the falloff as aspect ratio decreases
- Late efforts in slicing images to common dimensionality did not preserve assumption of consistent resolution and aspect
- Smaller objects clearly not captured





Conclusion

Conclusions

- Model usage dictates which model is preferable
  - Retinanet has substantially higher prediction accuracy (as measured by mAP) than Fast-YOLO
  - Fast-YOLO has a much faster inference speed than Retinanet, but much slower training time
- Understanding the input data and image resolution is crucial in evaluating model's ability to capture smaller objects
- Further customization to network architecture can likely address the issues in capturing objects of significantly different sizes