Object Detection From Satellite Images.

Hasin Md Daiyan
dept. of CSE
AUST
Dhaka, Bd.
hasin.cse.20200204055@aust.edu

Ahmed Al Nahian
dept. of CSE
AUST
Dhaka, Bd.
ahmed.cse.20200204063@aust.edu

Amit Karmakar
dept. of CSE
AUST
Dhaka, Bd
amit.cse.20200204056@aust.edu

The significance of object detection in aerial photographs has grown with the widespread use of satellite imaging technologies and unmanned aerial vehicles (UAVs). The study explores advanced techniques for object detection and classification in aerial photography, emphasizing the special difficulties caused by different scales, occlusions, and intricate backgrounds

***Index Terms*—YOLO,COCO**

I. Introduction

Object detection in Earth Vision refers to localizing objects of interest (e.g., vehicles, airplanes) on the earth's surface and predicting their categories[1]. As aerial images become more accessible and widely used, the need for effective object detection methods that can accurately identify and classify objects within these images is paramount. However, object detection in aerial imagery presents unique challenges due to factors such as diverse object scales, variations in lighting, atmospheric conditions, and the presence of occlusions caused by overlapping objects or complex terrains. Traditional object detection methods often struggle to adapt to these challenges, necessitating the development of advanced machine learning techniques. In particular, deep learning approaches, have shown remarkable promise in improving detection accuracy. This paper aims to provide a comprehensive overview of current methodologies in object detection for aerial images using model like yolo.

II. Literature Review

In recent years, aerial object detection has seen significant advancements with the development of large-scale datasets and improved deep learning models. The DOTA dataset, introduced by Gui-Song et al., provides 2806 high-resolution aerial images with 188,282 object instances across 15 categories, offering a more complex and realistic benchmark for object detection due to its dense and unbalanced object distribution [1]. Building on this, Bo Jiang et al. proposed VC-YOLO, a lightweight real-time object detection model designed for aerial images. This model incorporates a Receptive Field Extended Backbone (RFENet) for enhanced feature extraction and an improved Feature Pyramid Network (iFPN) to efficiently reuse features across convolution stages, demonstrating superior accuracy and efficiency on the NWPU VHR-10 and VisDrone datasets [2]. In a separate study, Atik et al. compared YOLOv2 and YOLOv3 on 43 aerial images across 9 object classes. YOLOv2 performed better in 5 of the 9 classes, with its best performance in detecting airplanes (99

III. Methodology

A. Dataset

DOTA is a large-scale dataset for object detection in aerial images. Each image is of the size in the range from 800 × 800 to 20,000 × 20,000 pixels. DOTA-v1.0 contains 15 common categories, 2,806 images and 188, 282 instances. The proportions of the training set, validation set, and testing set in DOTA-v1.0 are 1/2, 1/6, and 1/3, respectively. The object categories in DOTA-v1.0: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool. Annotation format : x1, y1, x2, y2, x3, y3, x4, y4, category, difficult

B. Pre-Processing

For the preprocessing part, the following steps were completed to prepare the dataset for object detection using YOLO:

- Conversion to COCO Format: Initially, the dataset was formatted into the COCO (Common Objects in Context) format. This included generating JSON files with the necessary annotations, such as image metadata, bounding box coordinates, and class labels, to align with the standard structure required for COCO-based object detection models.
- Custom Dataset Implementation: A custom dataset class was created to efficiently handle the aerial images and annotations. This class integrated the necessary functions for loading images, parsing the annotations, and applying data augmentations such as random flips and rescaling. This ensured that the dataset was compatible with different object detection models, including Faster R-CNN and YOLO.
- Conversion to YOLO Format: Since the YOLO model requires annotations in its own format, the dataset was

Fig. 1. COCO-ANNOTATION

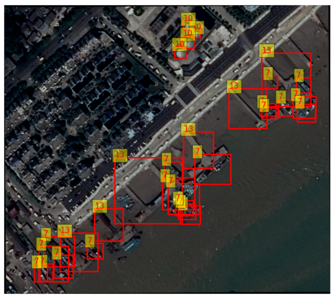

Fig. 2. Custom Dataset

further converted from COCO to YOLO format. This involved transforming the bounding box coordinates from the COCO system (which uses absolute pixel values) to the normalized format required by YOLO, where bounding boxes are expressed as relative values between 0 and 1. The annotations were saved as individual text files corresponding to each image, following YOLO's structure, with each line containing the class index and normalized bounding box coordinates.

C. Model

YOLO (You Only Look Once) is a popular object detection model. Training a YOLO model can be possible using the batches of images and annotations provided by the data loader. YOLO typically requires images and bounding box annotations (sometimes class labels) as inputs during training.YOLO is a single-shot detector that uses a fully convolutional neural network (CNN) to process an image. Faster R-CNN is another object detection model that uses region proposal networks to generate region proposals and then classifies and refines these proposals. Like YOLO, Faster R-CNN also requires images and bounding box annotations for training, which can be fed directly from the data loader.

IV. RESULT AND ANALYSIS

The YOLO object detection model achieved an overall precision of 68.3 percent recall of 19.3 percent, with mean average precision (mAP) values of 18.8 percent at IoU 0.5 and 10.8 percent across IoU thresholds from 0.5 to 0.95. Class-specific performance varied significantly, with the tennis court class performing well (precision: 74.6 percent, recall: 80.6 percent), while classes like bridge and helicopter exhibited low detection rates. The model's inference speed was efficient, processing images at 2.7 ms for preprocessing, resulting in a fitness score of 0.116.

Based on the results, the model can be considered below average. While it shows decent precision (68.3 percent), the low recall (19.3 percent) and mAP values (18.8 percent for mAP@50, 10.8 percent for mAP@50-95) indicate that it's missing a significant number of true positives. Additionally, the large variability in class performance—with some classes performing well (e.g., tennis court) but others poorly (e.g., bridge, helicopter)—suggests that the model struggles with consistent detection across object categories. The overall fitness score (0.116) further reflects the need for improvement.

CONCLUSION

In conclusion, the YOLO model demonstrates moderate precision but suffers from low recall and mean average precision, indicating a significant number of missed detections. While some classes like tennis court perform well, others such as bridge and helicopter exhibit poor detection rates, highlighting inconsistencies in the model's performance across object categories. The overall fitness score of 0.116 suggests that further improvements are needed, particularly through addressing class imbalance, enhancing data augmentation, and tuning model parameters for better accuracy.

REFERENCES

[1] Gui-Song,Xiang,Jian,Zhen,Serge,Jiebo,Mihai,Marcello,Liangpei - DOTA: A Large-scale Dataset for Object Detection in Aerial Images

[2] Bo Jiang,Ruokun Qu,Yandong Li and Chenglong Li-VC-YOLO: Towards Real-time Object Detection in Aerial Images

[3] Muhammed Enes ATIK, Zaide DURAN, Roni OZGUNLUK-Comparison of YOLO Versions for Object Detection from Aerial Images