

Finding Instances of Objects in Aerial Images using Earth Vision Small Scale Dataset

Neha Pawar¹, Ashwini Waghmare², Anuja Pratap³, Anjalee Thorat⁴, Kedar N. Ghogale⁵,
Shankar Naik Rathod Karamtoth⁶ and Nuzhat Faiz Shaikh⁷

^{1-4, 7}Modern Education Society's College of Engineering, Pune 411001, India

Email: nehapawar20001@gmail.com, ashwiniwaghmare6712@gmail.com,
anujapratap26329@gmail.com, anjaleet17@gmail.com, nfshaikh@mescoepune.org

⁵⁻⁶Centre For Development of Advanced Computing, Pune, India

Email: kedarg@cdac.in, kshankar@cdac.in

Abstract— Object detection has made enormous progress in naturalistic images during the last decade. However little has been done to detect miniscule objects in aerial images. Because of their small size and complex backstory detecting these small objects becomes very difficult. RCNN, Fast RCNN, Faster RCNN, YOLO, and SSD are some of the object detection algorithms utilized. Many changes have been made to these algorithms in order to enhance performance both in speed and accuracy. YOLOR i.e. You Only Learn One Representation is one such algorithm that has been developed specifically for object detection, unlike other algorithms which include object classification or object analysis. YOLOR is a one-stage detector, capable of performing numerous tasks at a time. In this paper, we present a brief summary of various object detection algorithms. We also present our dataset 'Earth Vision small-scale dataset' to make tiny object detection easier. The dataset consists of 700 high-resolution images. We also studied the performance of the YOLOR algorithm on this dataset. Experimental results show that the accuracy obtained is 76% which is good due to the small size of the dataset and the tiny nature of the objects. The source code of this work is at: <https://github.com/PawarNehaAshok/Yolor-EarthVision-Small-Scale-Dataset>.

Index Terms— EarthVision, small-scale dataset, miniscule, object detection, Yolor, implicit knowledge.

I. INTRODUCTION

Object Detection is a very vast field. There is much research going on in object detection in machine learning and deep learning. Among these 2 fields in this paper, we have focused on deep learning. Tiny object detection in aerial images is an open issue as the objects in the images are microscopic and due to that, it is difficult to detect the objects in the images. Due to the small size of the features extraction creating bounding boxes around objects is quite difficult, so we took up this challenge to detect such small objects in aerial images since it will be very useful in applications such as large-scale surveillance and intelligent transportation. While studying this issue we came across different algorithms which would help us detect the objects in aerial images such as SSD (Single Shot Detector) [1], YOLO (You only look once) [2], HOG (Histogram of Oriented Gradients) [3], SURF [4], SIFT and many more. But as per the latest studies in object detection, a new algorithm for object

detection was developed which was known as YOLOR (You only learn one representation). YOLOR was proposed as a “unified network to encode implicit knowledge and explicit knowledge together” [6]. Chien-Yao Wang, I-Hau, The, and Hong-Yuan Mark Liao are among the authors of the study You Only Learn One Representation: Unified Network for Multiple Tasks. It's not only used for object detection but also can be used for instance segmentation, panoptic segmentation, Image captioning. Why did we choose YOLOR for detecting tiny objects in aerial images? It's because its inference speed was increased by 88% than Scaled YOLO v4 and the mean average precision was increased by 3.8% compared to PP-YOLOv2. YOLOR (you only learn one representation) is different than Yolo (you look only once). Because of this methodology, YOLOR integrates implicit and explicit knowledge jointly, resulting in a refined general representation.

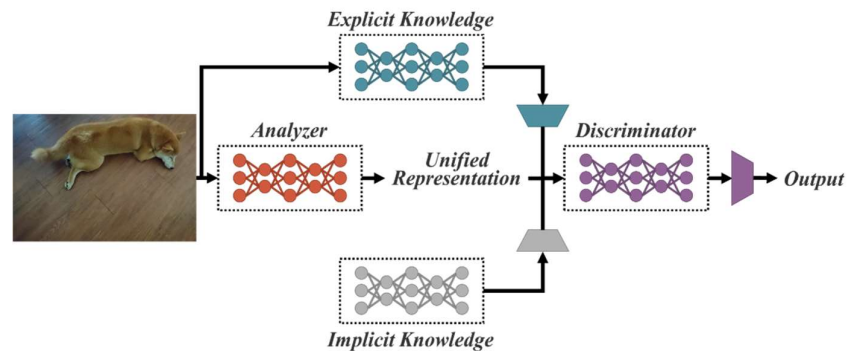


Fig. 1. YOLOR Architecture [6]

By supplying unambiguous information or image databases that are either properly annotated or well arranged, we provide explicit knowledge to the neural network. After that, the model becomes adept at classifying images into appropriate categories. While characteristics in deep layers provide implicit knowledge, the knowledge that does not correspond to observations is also known as implicit knowledge. YOLOR thus integrates implicit and explicit knowledge to create a general representation. CNN's are trained to fulfill one objective at a time whereas in YOLOR we can solve multiple problems at a single time [6], YOLOR tries to have CNN learn how to get output as well as what all different outputs could be. For this research, we have created our dataset consisting of various images from across the globe of 5 distinct categories, which is called Earth-Vision small-scale dataset. For creating the database, we collected images from Google earth which can be defined as aerial images. The images present in the dataset are such that it contains images with objects to be detected as well as images in which no objects are present so that our model is not over-fitted. Categories in which the data is divided are Airplanes, Bridges, playgrounds, Water-bodies, and Vehicles. Our goal is to detect all these categories if we provide images to them.

II. LITERATURE REVIEW

A. One Stage Detector

SSD (Single Shot Detector): A single-shot detector detects multiple objects in an image in one shot using a multi-box. It has higher accuracy and a faster speed. The architecture consists of VGG-16 layers and multi-box convolution layers. [1] VGG has extra convolutional layers which decrease in size at the end of the base these layers help in determining the features which are present in the bounding box. Hence it is able to detect objects on multiple scales better but it is not suitable for tiny object detection as it works worse for small scale objects.

YOLO (You Only Look Once): YOLO is short for You Only Look Once. The algorithm is famous for its accuracy and speed. It detects all objects in one run of the algorithm hence it is called as You Only Look Once. It divides the image into $S \times S$ grid. The grid in which center of an object lies is responsible for that object. Corresponding bounding boxes with different confidence scores are generated. The final bounding box is obtained through non max suppression. [2] When comparing the differences between a two-stage detector and a one-stage Detector, it is evident that the one-stage detector is the solution if the demand is Speed and how the faster system performs for that method. If accuracy is a pre-Requisite, the two-stage detector can be employed. YOLOv2, YOLOv3, YOLOv4, and YOLOv5 are some of the numerous variations of YOLO. In 2017, YOLOv2 was launched, and it received an honorable mention at CVPR 2017. BatchNorm, better resolution, and anchor boxes were among the incremental improvements made to the architecture on top of YOLO [7]. In 2018, YOLOv3 was

released. (Because of its conversational tone, the YOLOv3 paper is one of the most readable publications in computer vision research.) YOLOv3 improved performance on smaller objects by adding an objectness score to bounding box prediction, adding connections to the backbone network layers, and making predictions at three different degrees of granularity [8].

Alexey Bochkovskiy's YOLOv4, published in April 2020, was the first work in the "YOLO family" that was not written by Joseph Redmon. Improved feature aggregation, a "bag of freebies" (including augmentations), mish activation, and more were all introduced in YOLOv4 [9].

Glenn Jocher's YOLOv5 model, released in June 2020, is the first in the "YOLO family" to be released without a paper – and is, similarly, undergoing "ongoing development" on its repo. Glenn had been maintaining a PyTorch implementation of YOLOv3, but as he continued to develop the architecture, he decided to create a new repo called YOLOv5 [10].

YOLOR (You Only Learn One Representation): YOLOR creates a general representation of knowledge that takes advantage of both explicit and implicit knowledge. This process is achieved by integrating deep learning and compressive sensing. The YOLOR demonstrated equivalent object detection accuracy as the Scaled YOLOv4 when used in conjunction with state-of-the-art methodologies while increasing inference speed by 88 percent. [6] As a result, YOLOR is one of the most efficient object detection algorithms available in today's computer vision. At the same inference speed, YOLOR has a 3.8 percent greater mean average precision on the MS COCO dataset than PP-YOLOv2. Hence After seeing such a great result in the MS COCO dataset with YOLOR, we decided to go ahead with this algorithm and test our dataset 'EarthVision small scale dataset' on it.

B. Two-Stage Detector

R-CNN: Ross Girshick et al. [11] published a paper in 2014 that described a method for extracting only 2000 regions from an image using selective search, referred to as a region of interest (ROI). The R-CNN algorithm employs a two-stage detection procedure. The first stage identifies image regions that could potentially contain an object. In the second stage, each region's object is categorized. In the first module, RCNN creates category-independent region predictions. [11] The set of possible detections provided to the detector is defined by these region proposals. A convolutional network generates a fixed-length feature representation from each region in the second module. A collection of class-specific linear support vector machines makes up the third module. The R-CNN detector creates region predictions using an algorithm similar to Edge Boxes. The proposal parts have been cropped and scaled out of the image. CNN then classifies the reduced and reformatted regions. Finally, the region proposal anchor boxes are refined using a support vector machine (SVM) trained using CNN features. There are three different forms of R-CNNs. Each version tries to increase one or more of these processes' efficiency, speed, or effectiveness.

Fast-RCNN: In 2015, the same authors of the R-CNN algorithm solved the drawbacks of r-CNN and called that algorithm Fast-RCNN. So, the drawbacks of r-CNN were that we are extracting 2000 regions from each image and extracting features using CNN in the next step for each image 2000 times (because there are 2000 regions) which makes it very slow. So finally in this paper [12], the authors have stated that you don't have to feed every time 2000 regions proposals to the convolutional neural network. Instead, the convolutional A feature map is generated after the operation is completed once per image.

Faster-RCNN: Whether it is R-CNN or fast R-CNN, both of these algorithms use a selective search algorithm to get the region proposals. The selective search process is a very slow and time-consuming process. Therefore, Ren, Shaoqing, K. He, R. Girshick, and a. J. Sun, published a paper called "Faster R-CNN: towards real-time object detection with region proposal networks," [13]. In this paper, they have totally removed the selective search algorithm part to identify regions and used a new convolutional neural network to predict these regions' proposals and that is the main approach followed in faster R-CNN.

When Comparing the differences between a two-stage detector and one-stage detector, it is evident that the one-stage detector is the solution if the demand is speed and how faster system performs for that method. If accuracy is a prerequisite, the two-stage detector can be employed.

III. EXPERIMENTATION

A. Machine Configuration

Hardware Specifications:

- Micro Processor: Intel Xeon Silver 4216 with 16 Cores.
- RAM: 64GB
- Operating System: Ubuntu 18.04

- Graphic Card: NVIDIA Quadro P2200 5GB GDDR5
- HDD: 4000GB, 7200 RPM Speed
- SSD: 512GB

Software Specifications:

- Pycharm IDE
- Python 3.10.0
- Pytorch 1.10.0
- Mish-Cuda 11.1

B. Dataset

We have created our custom dataset, named as Earth Vision Small-Scale Dataset, to train the YOLOR (You Only Learn One Representation) model. The images for the dataset are taken from Google Earth App. In a 70:20:10 ratio, the dataset was divided into training, validation, and testing. It has a total of 700 images. Our dataset has a total of five categories airplane, bridge, playground, vehicles, and water-bodies. For annotations, we have used the Roboflow framework. Roboflow is the go-to tool for converting computer vision datasets. You can use the annotate tool to annotate your images after uploading them to roboflow. Any annotation format can be imported, and any format can be exported. We used roboflow's auto-orient and resize it to 416x416 pixels for preprocessing dataset images.

The dataset is available on Roboflow which is EarthVision small-scale dataset version 4. It contains annotated images for the training and validation set. Dataset can be downloaded using following URL (<https://app.roboflow.com/ds/o7vfc5ODBD?key=eVYmePCUq3>).



Fig. 2. Samples of testing images in the Earth-vision small scale dataset

IV. RESULTS

The YOLOR model is pre-trained on the MSCOCO dataset already. Based on research, it has been concluded that in YOLOR they have used explicit knowledge with implicit knowledge (conscious learning) which shows the competitive results for object detection problems. So by getting inspired by these results, we have also trained the YOLOR P6 model on our Earth Vision Small-Scale Dataset.

We employed transfer learning to train YOLOR. Transfer learning works on the simple principle of taking a model that has been trained on a large dataset and applying it to a smaller dataset. The training is completed on the YOLOR P6 model by keeping, a batch size of 10 and 300 epochs. The image size is 448. For training YOLOR we have used CPU but it can also be done on GPU. Precision means how accurate your predictions are. Recall means how good your model is at finding all positives. Mean Average Precision is nothing but the

combination of every result to create one value that will judge the object detection. And the formula that is used to calculate results is below-

TP - True Positive (i.e. the model correctly predicts the correct class) [14], TN - True Negative (i.e. the model correctly predicts the wrong class) [14], FP - False Positive (i.e. the model incorrectly predicts the correct class) [14], FN - False Negative (i.e. the model incorrectly predicts the wrong class) [14]

Precision = $TP / (TP + FP)$ [15]

Recall = $TP / (TP + FN)$ [15]

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$ [15]

We have computed confusion matrix which shows the precision, recall and accuracy of each class. So, the first class is Airplane.

Airplane

TABLE I. CONFUSION MATRIX FOR AIRPLANE

N=18 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	14(TP)	3(FN)	17
Actual Negative	1(FP)	0(TN)	1
Total	15	3	18

From the above matrix, Precision (Airplane) = 93%, Recall (Airplane) = 82%, Accuracy (Airplane) = 77%

Bridge

TABLE II. CONFUSION MATRIX FOR BRIDGE

N=17 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	10(TP)	2(FN)	12
Actual Negative	2(FP)	3(TN)	5
Total	12	5	17

From the above matrix, Precision (Bridge) = 83%, Recall (Bridge) = 83%, Accuracy (Bridge) = 76%

Playground

TABLE III. CONFUSION MATRIX FOR PLAYGROUND

N=23 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	16(TP)	0(FN)	16
Actual Negative	7(FP)	0(TN)	7
Total	23	0	23

From the above matrix, Precision (Playground) = 69%, Recall (Playground) = 100%, Accuracy (Playground) = 69%

Vehicle

TABLE IV. CONFUSION MATRIX FOR VEHICLE

N=18 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	15(TP)	3(FN)	16
Actual Negative	0(FP)	0(TN)	7
Total	15	3	18

From the above matrix, Precision (Vehicle) = 100%, Recall (Vehicle) = 83%, Accuracy (Vehicle) = 83%

Water-Bodies

TABLE V. CONFUSION MATRIX FOR WATER-BODIES

N=25 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	16(TP)	2(FN)	18
Actual Negative	5(FP)	2(TN)	7
Total	21	4	25

From the above matrix, Precision (Water-Bodies) = 76%, Recall (Water-Bodies) = 80%, Accuracy (Water-Bodies) = 72%

Since we have seen the Precision, Recall, and Accuracy of each class respectively so now we will see the above factors for all categories combined in a confusion matrix.

TABLE VI. CONFUSION MATRIX FOR ALL CATEGORIES (AIRPLANE, BRIDGE, PLAYGROUND, VEHICLE AND WATER-BODIES)

N=85 samples	Predicated Positive	Predicated Negative	Total
Actual Positive	60(TP)	9(FN)	69
Actual Negative	11(FP)	5(TN)	16
Total	71	14	85

From the above matrix, Precision = 84%, Recall = 86%, Accuracy = 76%

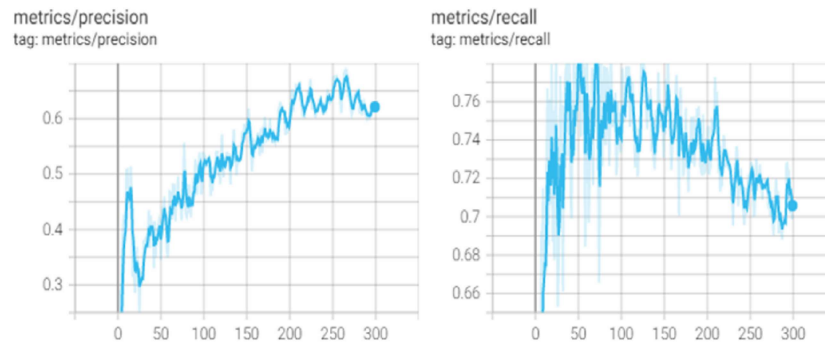


Fig. 3. Training graphs (part I) (on x-axis = epochs, on y-axis = percentage)

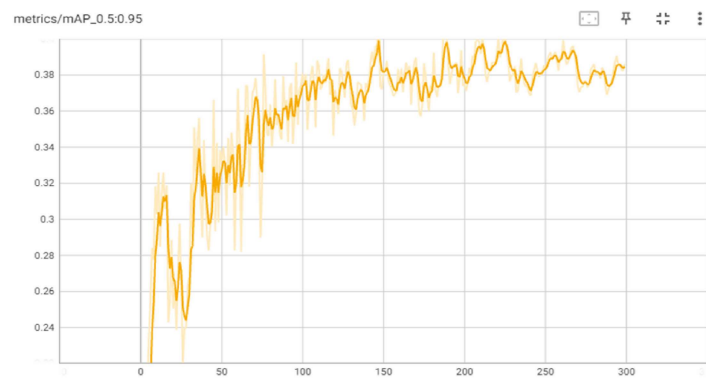


Fig. 4. Training graphs (part II) (on x-axis = epochs, on y-axis = percentage)

V. CONCLUSION

In this paper, we have introduced our dataset 'EarthVision small-scale dataset' for detecting miniscule objects in general and vehicles, bridges, airplane, playgrounds and water bodies in particular. The YOLOR algorithm is extensively used today for object detection. Findings reveal that the YOLOR algorithm outperforms various other algorithms by increasing the localization performance of miniscule object detection. We examined the performance of YOLOR model on our EarthVision small-scale dataset. The overall accuracy obtained is 76%. This result is attributed to the small size of dataset and tiny nature of objects. Tiny object detection related work is of immense importance today considering its applications in various fields. The progress in this field will prove to be helpful in many computer vision applications involving object detection in the future.

REFERENCES

- [1] Liu, W. a. Anguelov, D. a. Erhan, D. a. Szegedy, C. a. Reed, S. a. Fu, C.-Y. a. Berg and A. C, "Ssd: Single shot multibox detector," in European conference on computer vision, USA, 2016.
- [2] Redmon, Joseph, S. Divvala, R. Girshick and a. A. Farhadi, "You only look once: Unified, real-time object detection," in In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
- [3] Dalal, N. a. Triggs and Bill, "Histograms of oriented gradients for human detection," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), France, 2005.
- [4] Bay, H. a. Tuytelaars, T. a. V. Gool and Luc, "Surf: Speeded up robust features," in European conference on computer vision, 2006.
- [5] F. A. a. W. Z. a. F. D. D. Pavel, "Reliable object recognition using sift features," in 2009 IEEE International Workshop on Multimedia Signal Processing (pp. 1-6). IEEE., October 2009.
- [6] C. Y. Y. I. H. & L. H. Y. M. Wang, "You only learn one representation: Unified network for multiple tasks.," in arXiv preprint arXiv:2105.04206, 2021.
- [7] Y. a. Q. J. a. H. Y. Wei, "Aerial image location of unmanned aerial vehicle based on yolo v2," in Laser & Optoelectronics Progress, 2017.
- [8] L. a. L. S. Zhao, "Object detection algorithm based on improved YOLOv3," in Multidisciplinary Digital Publishing Institute, 2020.
- [9] Bochkovskiy, A. a. Wang, C.-Y. a. Liao and H.-Y. Mark, "Yolov4: Optimal speed and accuracy of object detection," in arXiv preprint arXiv:2004.10934, 2020.
- [10] D. Thuan, "Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm," 2021.
- [11] R. a. D. J. a. D. T. a. M. J. Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.
- [12] Girshick and Ross, "Fast r-cnn," in In Proceedings of the IEEE international conference on computer vision, 2015.
- [13] Ren, Shaoqing, K. He, R. Girshick and a. J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," IEEE transactions on pattern analysis and machine intelligence, vol. 39, pp. 1137--1149, 2016.
- [14] S. Soo, "Object detection using Haar-cascade Classifier," in Institute of Computer Science, University of Tartu, 2014.
- [15] R. a. N. S. L. a. D. S. E. A. Padilla, "A survey on performance metrics for object-detection algorithms," in 2020 international conference on systems, signals and image processing (IWSSIP), 2020.