Spatial Vehicle Detection using Convolutional Neural Networks

Carlo Longhi
carlo.longhi@studio.unibo.it

Alma Mater Studiorum Università di Bologna

February 6, 2023

Abstract

This work deals with the problem of vehicle detection in aerial images. It makes use of a challenging dataset found on Kaggle made of satellite images acquired from Google Earth Pro. Two different models are employed: a FasterRCNN and a model from the YOLOv5 family. The results obtained are promising but underline the difficulty of training these models on aerial images in a class-unbalanced scenario.

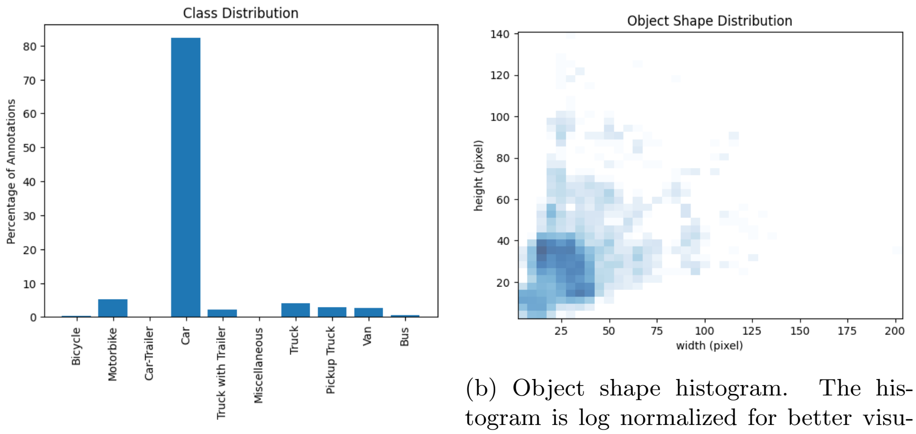
1 Introduction

Object detection is one of the fundamental tasks in the field of computer vision and various models based on Convolutional Neural Networks (CNNs) have been devised to solve it [10] [9] [6] [2]. Promising results have been achieved in different datasets like MS COCO [5] and VOC2007 [1]. However, aerial images pose a different set of challenges that need to be dealt with, like the small size of the objects and their cluttered arrangement. Moreover, detecting objects in aerial images is an important task in many practical applications, for example resource detection, environmental monitoring and urban planning [11].

In this work we focus on the task of vehicle detection and use a dataset called Spatial Vehicle Detection [7], made of satellite images collected in an urban scenario.

2 Data Analysis

The Spatial Vehicle Detection dataset is made of 100 images of satellite data acquired from Google Earth Pro in the area of Edogawa, Tokyo in Japan. The vehicles present in each image are annotated with bounding boxes and are assigned one of these 10 classes: *Bicycle, Motorbike, Car-Trailer, Car, Truck with*



(a) Class distribution in the dataset.

(b) Object shape histogram. The histogram is log normalized for better visualization.

Trailer, Miscellaneous, Truck, Pickup Truck, Van, Bus. The dataset contains 17609 annotations of vehicles and its main characteristics are:

1. The images are big compared to other Computer Vision datasets: the average dimension is 1834x887 pixels.
2. Highly unbalanced object annotations. Figure 1a shows the class distribution.
3. Majority of small objects with some outliers. See Figure 1b.

3 Experimental Setup

Two different models are employed: FasterRCNN [10] and YOLOv5 [4]. The dataset is split in training, validation and test set using respectively 70%, 10% and 20% of the original dataset.

To reduce the memory needed during the training process, given the large dimension of the images, we divide them in 4 taking the top-left, top-right, bottom-left and bottom-right crops. To avoid missing annotations in the border regions, the crops are increased by 200 pixels in each dimension making them partially overlapping.

3.1 FasterRCNN

A FasterRCNN model with a Resnet-50 [3] backbone was used for our experiments. The backbone used is pretrained on the ImageNet1K dataset. The model is trained for 50 epochs with a learning rate equal to $5 \cdot 10^{-5}$ using early stopping with a patience of 20 epochs and the Adam optimizer. To prevent

Feature Level	Anchor Size
1	4,6,8
2	8,10,12
3	16,20,24
4	32,40,50
5	64,80,128

Table 1: Anchor sizes used at the relative feature level.

the model overfitting early in the training process, a set of image augmentations where implemented. The augmentations include random horizontal and vertical flipping and color jitter.

To exploit the higher spatial resolution of low level features, all 5 level of features produced by the backbone are used by the Region Proposal Network. At each feature level, 9 anchors are used by combining 3 aspect ratios $0.5, 1, 2$ and 3 different anchor sizes. The anchor sizes are reported in Table 1.

3.2 YOLOv5

YOLOv5 is a family of compound-scaled object detection models that include simple functionalities for training and hyperparameters tuning [4]. The model chosen for our task is YOLOv5l. We use weights pretrained on the COCO dataset and image augmentations acting on hue, saturation, scale, horizontal flipping and image mosaic. In our tests the batch size is set to 4 and the SGD optimizer is used to train the model for 600 epochs. The anchors used by YOLO are determined automatically by first applying a *k-means* function to the dataset labels, then the anchor centroids found in this way are used as initial conditions for a Genetic Evolution algorithm. The resulting anchors are used by our model.

4 Results

The performance of the models are measured using the COCO mAP metric. Results obtained with the FasterRCNN model are reported in Table 2, with the best performing model achieving a mAP equal to 0.13 on the test set. The results obtained by the best model on each single class are shown in Table 3.

YOLOv5 performs better than FasterRCNN on this dataset, reaching a mAP on the test set of 0.164. The mAP values for each class are reported in Table 4.

The underwhelming results obtained for the objects of class *Bicycle* or *Motorbike* reflect the difficulty of detecting smaller objects. The dimensions of their bounding boxes, shown in Figure 2 pose a challenge for our trained models. Moreover, our models are unable to detect objects from very infrequent classes such as *Car-Trailer* and *Miscellaneous*. *Car-Trailer* annotations are less than the 0.02% of the total dataset objects and *Miscellaneous* less than the 0.1%. On the other hand both models obtain good performances on the most frequent class of objects *Car*, representing the 81% of all samples.

	Validation mAP	Test mAP
FasterRCNN	0.034	0.036
+ pretrained backbone	0.09	0.092
+ image augmentations	0.096	0.114
+ all feature layers	0.111	0.13

Table 2: FasterRCNN results

Class	Val mAP	Class	Val mAP
Bicycle	0	Miscellaneous	0
Motorbike	0.007	Truck	0.212
Car-Trailer	0	Pickup Truck	0.154
Car	0.401	Van	0.139
Truck with Trailer	0.24	Bus	0.148

Table 3: COCO mAP values for each class of the test set using FasterRCNN.

Class	Val mAP	Class	Val mAP
Bicycle	0	Miscellaneous	0
Motorbike	0	Truck	0.36
Car-Trailer	0	Pickup Truck	0.137
Car	0.636	Van	0.03
Truck with Trailer	0.463	Bus	0.009

Table 4: COCO mAP values for each class of the test set using YOLOv5.

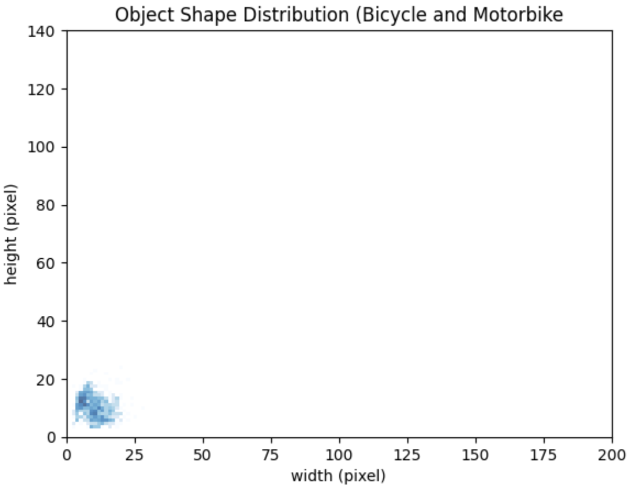


Figure 2: Object shape histogram fot the Bicycle and Motorbike classes.

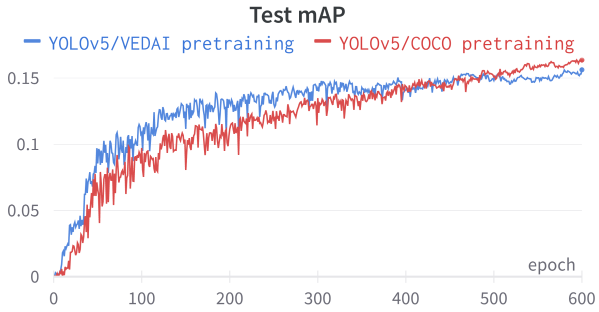


Figure 3: Test mAP obtained training YOLOv5 using COCO or VEDAI-pretrained weights.

4.1 Pretraining

To draw the full potential of using pretrained weights, we train our models on another dataset containing aerial images. The dataset used is Vehicle Detection in Aerial Images (VEDAI) [8], a dataset for object detection composed of 1210 1024x1024 images with annotation for 12 classes of vehicles: *car, truck, pickup, tractor, camping car, boat, motorcycle, bus, van, other, small, large*. Both models were trained on the VEDAI dataset using the same hyperparameters described above. Our FasterRCNN model obtains a mAP of 0.232 on the VEDAI validation set and YOLOv5 outperforms it with a mAP of 0.43.

The pretrained weights are then used to initialize the models before training on the smaller vehicle detection dataset. Our FasterRCNN finetuned on the Spatial Vehicle Detection dataset obtains a mAP of 0.138, outperforming all the previous configurations. YOLOv5 pretrained on VEDAI achieves a slightly lower mAP after finetuning than our previous tests. On the other hand, its training process gets better results faster as shown in Figure 3.

5 Conclusions

The results highlight the challenges posed by the chosen dataset and the vehicle detection task in aerial images. The task of object detection becomes very challenging when dealing with such small objects and highly unbalanced classes. Further work to improve on these results should take these complications in consideration.

After some tests, it is clear that the models of the YOLOv5 family are really easy to use and can obtain very promising results with little to no parameter tuning.

References

[1] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**, 303–308 (2009)

[2] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)

[3] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[4] Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomammana, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, J., Fang, J., , L.Y., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, Rai, P.: ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (Oct 2020). https://doi.org/10.5281/zenodo.4154370, `https://doi.org/10.5281/zenodo.4154370`

[5] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)

[7] Ltd., A.A., Roomy, S., Nayem, A.B.S., Tonmoy, A.M., Islam, S.M.S., Islam, M.M.: Spatial vehicle detection (2022). https://doi.org/10.34740/KAGGLE/DSV/4449798, `https://www.kaggle.com/dsv/4449798`

[8] Razakarivony, S., Jurie, F.: Vehicle detection in aerial imagery: A small target detection benchmark. Journal of Visual Communication and Image Representation **34**, 187–203 (2016)

[9] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)

[10] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. p. 91–99. NIPS'15, MIT Press, Cambridge, MA, USA (2015)

[11] Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3974–3983 (2018)