Technische Universität Berlin
MSc: Geodesy and Geoinformation Science
Dept: Computer Vision & Remote Sensing

Aerial Video Understanding by using Object Detection methods

Project Hot Topics in Computer Vision:
Video Understanding

Emmanouil Papadakis (M.N.: 413030)
Stylianos Kossieris (M.N.: 409504)

Prof.: Olaf Hellwich

Berlin, July 2021

1. Introduction

Based on the use of SIFT [i] and HOG [ii], many various visual recognition tasks has been developed the last decade. However, the deep convolutional networks have outperformed the state of the art algorithms in many visual recognition tasks. Convolutional Networks have already existed for a long time but their success was limited due to the size of the available training sets and the size of the considered networks. The typical use of Convolutional Networks was on classification tasks, where the output to an image is a single class label. In 2012, the breakthrough by Krizhevsky et al. was due to supervised training of a large network with millions of parameters and 8 layers on the ImageNet dataset with 1 million training images [iii]. However, in many computer vision tasks, the desired output should include localization, i.e., a class label is supposed to be assigned to one smaller area –bounding box recognition- of the image (Object Detection) or to each pixel (Segmentation).

In order to find the most suitable Object Detection Algorithm for the understanding of aerial video, we search between the benchmarks Object Detection algorithms/ Neural Networks. The first category is a combination of region proposals with CNNs, as R-CNN [ix], Fast R-CNN [x] and Faster R-CNN [xi]. The algorithms extract a constant number of region proposals per image, which for R-CNN is 2000 and for Faster R-CNN is 300. In R-CNN a set of linear SVMs score each feature vector, while in Fast-CNN there are two output vectors per Region of Interest (RoI): Softmax Probabilities and Per-class bounding-box offsets. In the latter, the classifier is run at all scales of pyramids of images and feature maps which are built. As result, the training of these algorithm is a multi-space pipeline and expensive in space and time.

On the other hand, single-shot object detectors were developed for multiple categories with high improvement in training speed and detection accuracy. Single Shot MultiBox Detector –SSD [xiii] and You Only Look Once –YOLO [vi] are methods which encapsulates all computation in a single deep neural network since the above completely eliminate proposal generation and subsequent pixel or feature resampling stages. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. SSD achieved high improvement in training speed and detection accuracy compared to YOLOv1 since, on Pascal VOC 2007 dataset, from 63.4% mAP of YOLOv1 succeeded 74.3% mAP.

Since our target was the training of a Neural Network for the detection of bicyclists, pedestrians, skateboarders, carts, cars and busses from an aerial video, the state-of-the-art algorithm YOLOv4 [iv] was the best choice. According to Bochkovskiy et al. , YOLOv4 is the most accurate object detector for recognizing multiple small-sized objects in images. In our training dataset, the videos

are captured with 1400x1080 resolution and the objects in images are depicted in areas which cover around 30x30 pixels. YOLOv4 is superior to the fastest and most accurate detectors in terms of both speed and accuracy. YOLOv4 except for Backbones, which is usually pre-trained in ImageNet, and the head, which is used for prediction of classes and bounding boxes, uses Neck in order to add some more layers. In object detection, multiple bounding boxes need to be drawn around images along with classification, so the feature layers of the convolutional backbone need to be mixed and held up in light of one another. The combination of backbone feature layers happens in the neck while Detection happens in the head. The optimal model, as the backbone of YOLOv4 detector, is the CSPDDarknet53 neural network. Moreover, YOLOv4 uses bag-of-freebies (BoF) in order to increase the variability of the input images but without growing the inference cost. Moreover, YOLOv4 uses bag-of-specials (BoS) enhancing certain attributes in a model, such as enlarging receptive field, introducing attention mechanism and strengthening feature integration capability. Bochkovskiy et al. proved that adding BoF and BoS training strategies, the mini-batch size has almost no effect on the detector's performance. As a result, after the introduction of BoF and BoS, it is no longer necessary to use expensive GPUs for training. As it can be understood, anyone can use only a conventional GPU to train an excellent detector.

2. Stanford Drone Dataset

Stanford Drone Dataset [xvii], was created for tasks such as target tracking and trajectory forecasting [v]. It is a very large scale dataset that navigates in a real world outdoor environment such as a university campus. As it is depicted below, the dataset collects images and videos of six various types of agents, these are: pedestrians, bicyclists, cars, skateboarders, golf carts and buses. The dataset consists of eight unique scenes and occupies 69 GB data storage. The number of videos in each scene and the percentage of each agent in each scene are reported below:

Scenes	Videos	Bicyclist	Pedestrian	Skateboarder	Cart	Car	Bus
gates	9	51.94	43.36	2.55	0.29	1.08	0.78
little	4	56.04	42.46	0.67	0	0.17	0.67
nexus	12	4.22	64.02	0.60	0.40	29.51	1.25
coupa	4	18.89	80.61	0.17	0.17	0.17	0
bookstore	7	32.89	63.94	1.63	0.34	0.83	0.37
deathCircle	5	56.30	33.13	2.33	3.10	4.71	0.42
quad	4	12.50	87.50	0	0	0	0
hyang	15	27.68	70.01	1.29	0.43	0.50	0.09

Table 1: Different scenes and Percentage of each agent in each scene

As it can be seen, there is data imbalance between different classes, since the percentage of bicyclists and pedestrians in each scene is by far bigger than the percentages in the other four classes. This data imbalance could be cause problems during the training of the algorithm. However, bag-of-freebies methods are dedicated to solving the very important issue of data imbalance between different classes. Lin et al. [xv] proposed focal loss to deal with the problem of data imbalance existing between various classes. As it can be understood, this issue of data imbalance played a major role in the choice of YOLOv4 algorithm.

Data collected at peak hours and each scene is captured with a 4K camera mounted on a quad copter platform flying at around 80m height. As it is depicted in Table 2, the total time of 60 videos is 4 hours and 49 minutes.

Scenes	Videos	Total time of Videos
bookstore	7	55min 54sec
coupa	4	26min 36sec
Deathcircle	5	22min 18sec
gates	9	26min 3sec
hyang	15	71min
little	4	24min 28sec
nexus	12	62min
quad	4	1min 4sec
Total amount:	**60**	**4h 49min**

Table 2: Scenes and Total time of Videos

For example, two scenes of the dataset Bookstore and Hyang area are depicted below respectively:

Figure 1: Frame of Bookstore Area

Figure 2: Frame of Hyang Area

In the following images, red bounding boxes (bicyclist targets) and magenta bounding boxes (pedestrian targets) target the detected objects by the algorithm.

Figure 3: Mini roundabout in campus

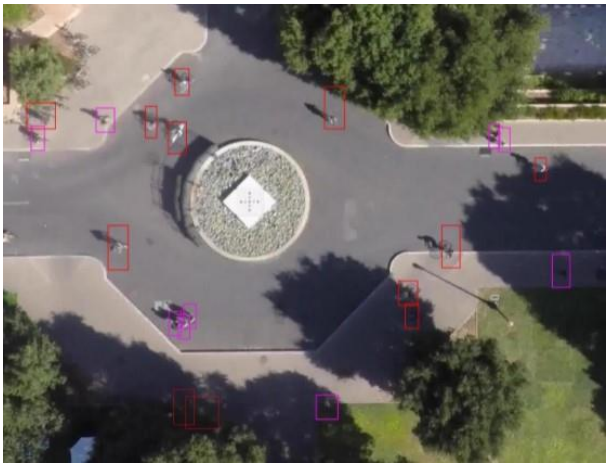

Figure 4: Pedestrian crossing in campus

3. Pretrained Model and Comparison

All YOLO models are object detection models. Object detection models are trained to look at an image and search for a subset of object classes. When found, these object classes are enclosed in a bounding box and their class is identified. Object detection models are typically trained and evaluated on the COCO dataset which contains a broad range of 91 object categories [xiv]. If object detection models are exposed to new training data, they will generalize to new object detection tasks.

Using weights of pretrained models, reassures the validity of the model in given images. As it can be seen in below images, the model works pretty well in images which are similar to the images of MS COCO dataset, in which the weight's model have been trained. On the other hand, it is really hard to identify the different object categories in aerial images of Stanford Drone Dataset, which are completely different than the images of MS COCO dataset.

Figure 5: Detection of Emmanouil, dog and cars with high accuracy

Figure 6: Detection of people in Acropolis area

As shown, the model encloses the correct object classes in bounding boxes accurately for people, cars, road sign and dog.

On the other hand, the algorithm does not recognize any object in Bookstore and Hyang scenes respectively. In the third image, of Deathcircle scene, the algorithm encloses two bicyclists in bounding boxes but it predicts wrong object class (motorcycle). Also, in the last image, the model recognizes two tables as clocks because of the shape and pattern. This is normal as it is not trained with aerial images, so it cannot detect and identify the objects of these images.

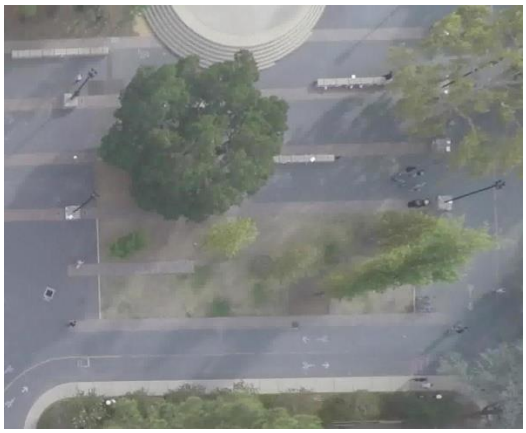

Figure 7: Impossible detection of object classes

Figure 8: Impossible detection of the object classes

Figure 9: False Detection of two bikers

Figure 10: False Detection of two tables

As result, it can be understood that for the identification of the correct object class in aerial images, the model is necessary to be trained using video of the Stanford Drone Dataset.

4. Preprocessing of Data

Since the amount of data occupies huge storage and the use of the whole dataset is unnecessary and time-consuming, it was decided to train the algorithm with the use of 3940 frames. Firstly, videos were cut in frames (3191) which is the whole dataset set. The dataset was split in training set that is used to train the model, test set that is used for testing the results of training and last the validation set in order to validate the final results. The size of each set according to bibliography should be: 80% for training set and 10% for test and validation set. For the training of algorithm were used images from all the different areas of dataset, except for Nexus area, in order to be used for the test of trained model.

Frames are accompanied by a .csv file which includes: the name of each image, coordinates of the bounding box (top left corner and bottom right corner) and the class of each object. However, YOLOv4 takes this information as .txt including the class, top left coordinates x, y and width, height of bounding box (***J. Redmon et al, 2016***). As a result, for each image a .txt file was created which includes the above information, normalized to each image.

Initially, the frames are necessary to have common dimensions. So, the minimum dimensions of frames are found among the dataset (width = 1322, height = 848) and then all the images are resized to these minimum width and height. Also, for the implementation of YOLOv4 algorithm, the normalization of coordinates of annotations to new common frame dimensions. The new coordinates are calculated as it can be seen below:

$$coord1 = \frac{\left(x * min_{width} / width\right)}{min_{width}} \qquad coord2 = \frac{\left(y * min_{height} / height\right)}{min_{height}}$$

$$coord3 = \frac{\left(w * min_{width} / width\right)}{min_{width}} \qquad coord4 = \frac{\left(h * min_{height} / height\right)}{min_{height}}$$

Moreover, a configuration file, ssd.yaml, was created with the paths of each set, the number of classes (6) and the names of classes in correct form (pedestrian, biker, car, skater, cart and bus). All python scripts will be attached in the deliverable repository.

5. Training of the model

The numerous studies of [iv] demonstrate that the CSPResNext50 is considerably better compared to CSPDarknet53 in terms of object classification on the ILSVRC2012 (ImageNet) dataset [iii]. Conversely, the CSPDarknet53 is better compared to CSPResNext50 in terms of detecting objects on the MS COCO dataset. CSPDarknet53 contains 29 convolutional layers 3x3, a 725 x 725 receptive field and 27.6 M parameters. Numerous experiments by Bochkovskiy et al. [iv] prove that CSPDarknet53 architecture is the optimal of the two as the backbone for a detector.

After investigation, for the initial training of YOLOv4 algorithm, there are not used initial pretrained weights, since as described in chapter 3, are originated from the training of algorithm in Coco dataset, which consists of ground and not aerial images. So, this kind of weights would

not help in a faster and more accurate training of the algorithm. Furthermore, mish cuda activation function was replaced by ReLU activation function because mish cuda is more complicated to run with docker and has no significant impact if ReLU activation function is used. The number of epochs selected 300, while the batch size equal to 16. Also, the dimensions of input frames selected to be 640x640. Because of lack of GPU memory in conventional laptops, the training process could not start. So, batch size and nominal batch size has to be reduced to 1 and the input frames size to 320x320 in order to train the model. Although these parameters minimized to lowest values, the conventional GPU MX130 with 2GB memory did not succeed to complete even the 1st training epoch. The solution was the use of a server with greater GPU memory.

This is Umbriel server of T.U.B., which uses as operating system Ubuntu 18.04 and consists of a GPU RTX 2070 xx (8GB VRAM) and RAM of 16 GB. The procedure ran with docker container [xvii].

6. Capturing video by UAV

The new aerial video by UAV captured in order to inference our model in a new completely different area than Stanford Drone Dataset. The capture of video was achieved by DJI Phantom 4 UAV in the city of Athens in Greece. As it is explained in chapter 1, YOLO struggles to precisely localize really small objects in images. So, the selected UAV flew in height around 40 meters, in lower height than the 80 meters of flight of Stanford Drone Dataset. A frame of the new aerial video is shown below:

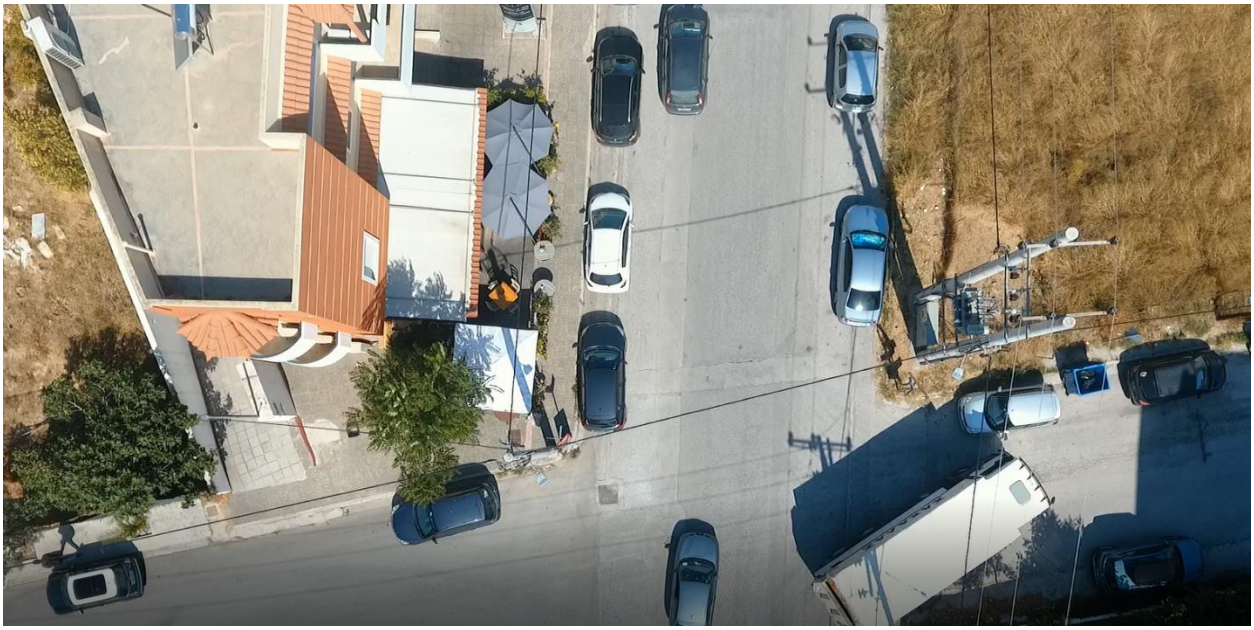

Figure 11: Frame of Video from area of Athens, Greece

As it is illustrated in the image, cars, bus and pedestrian are depicted in that frame of video.

Apart from the inference of model in our own aerial video, the inference of model in Nexus area is going to take place, as frames of Nexus area don't take part in training procedure. We are waiting higher accuracy during the inference of model in Nexus area than in the area of Athens, since frames have common characteristics with the other six areas of Stanford Dataset. The video UAVvideo.mp4 can be found in repository.

7. Conclusions

Comparison of the results obtained with other state of- the-art object detectors, YOLOv4 are located on the Pareto optimality curve and are superior to the fastest and most accurate detectors in terms of both speed and accuracy. The detector used can be trained on a conventional GPU with 8-16 GB-VRAM making its broad use possible. The original concept of one-stage anchor-based detectors has proven its viability.

During the processing of repository a lot of problems faced in order to make the code compatible with different systems (conventional laptops and server). Training process worked but due to the lack of GPU memory, the training process could not run.

Next steps would be to take the weights of training model and the assessment of the accuracy of bounding boxes. In order to evaluate the accuracy of the model AP (Average Precision) and mAP (mean Average Precision) indicators are used. Also, the curve of precision and recall shows the results of the algorithm. What it would be expected is to see bounding boxes to the frames of the new video and high rates to AP and mAP indicators.

Finally, the video (UAVvideo.mp4 in repo) which is the evaluation video of training process has to be cut in frames and inserted to the model with output weights from training.

References

i. D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.

ii. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

iii. A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

iv. A. Bochkovskiy, C.Y Wang, H.Y.M Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. (CVPR) (2020)

v. A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese. Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes. European Conference on Computer Vision (ECCV) (2016)

vi. J. Redmon, S. Divvala, R. Girshick, A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. (CVPR) (2016)

vii. K. He, G. Gkioxari, P. Dollár, R. Girshick. Mask R-CNN (CVPR) (2018)

viii. K. Vasili. Object Detection from High Resolution Aerial Video Data with CNNs. Athens (2018)

ix. R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. (CVPR) (2014)

x. R. Girshick Fast R-CNN (CVPR) (2015)

xi. S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (CVPR) (2016)

xii. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T. *et al.* Selective Search for Object Recognition. *Int J Comput Vis* 104, 154–171 (2013).

xiii. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg. SSD: Single Shot MultiBox Detector (CVPR) (2016)

xiv. T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollar Microsoft COCO: Common Objects in Context (CVPR) (2015)

xv. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Doll´ar. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, 2017.

xvi. https://pjreddie.com/darknet/yolo/

xvii. https://cvgl.stanford.edu/projects/uav_data/

xviii. https://hub.docker.com/r/pytorch/pytorch/tags?page=1&ordering=last_updated