The relation between object area and accuracy for a modern mobile convolutional object detector

Matthijs Rijlaarsdam
TU Delft

mscrijlaarsdam@student.tudelft.nl

Abstract

Object detectors, much like humans, perform less well on small than on large objects. Because of this, the object size distribution of a dataset influences the average precision a network achieves on that dataset. Therefore, the object size/precision curve of a network might be a better way to compare convolutional object detectors than the average precision over an entire dataset. In this paper we measure the relationship between object size and accuracy for a modern mobile convolutional object detector. We verify that this relationship holds for a different dataset, and that the dataset object size distribution influences the average precision over the entire dataset. We conclude that the object size/accuracy curve might contain more information about a network's performance than the average precision over an entire dataset.

1. Introduction

The performance of convolutional object detectors has been improving both in terms of precision and inference speed in recent years. R-CNN [1] introduced region proposals using selective search and made convolutional neural nets computationally feasible for object detection by applying convolutions on regions of interest. Fast R-CNN [2] improved on this idea by generating a feature map of the entire image. Faster R-CNN [3] sped up detections by generating the region proposals using a region proposal network. With the advent of single-pass techniques such as SSD [4] and YOLO [5], object detectors are becoming feasible for mobile devices.

When selecting a network for a given task, parameters such as network precision, size and speed have to be taken into account. For instance, a mobile application that detects cars in real-time might require a smaller and faster network than one detecting anomalies in x-rays, which needs more precision.

When comparing network precision, the (mean) average precision (mAP or AP) on benchmark datasets such as COCO [6] is usually used. However, object detectors, much like humans, perform better on large objects than on small objects. The average precision over an entire dataset is therefore influenced by the object size distribution of the dataset used. Depending on how much the performance of a network varies with object size, using the average precision for an entire dataset as a performance metric might leave out information about the performance of a network.

An object detector with a low AP on for instance the COCO dataset (which has a lot of small objects) might have a high precision on larger objects. Therefore, measuring the relationship between object size and precision can aid in deciding what model to use. This is especially relevant for mobile applications, where inference speed and model size are more important. For example, when selecting a network to detect pedestrians for a self driving car using a mobile platform, a network that performs in real time and has a high precision on large objects might be a better choice than one that is slow but has state-of-the art precision on COCO. We illustrate this idea in figure 1.

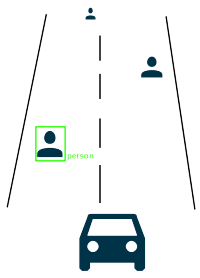

Figure 1. Object detectors work better on larger objects than small ones. While the object detector in a self driving car might detect 30% of all people it sees, it might detect 100% of people closer than 10 meters.

In this paper we examine the relationship between ob-

ject size and average precision for a modern mobile convolutional object detector[1].

Our main contributions are as follows:

- We show that there is significant relationship between object size and precision for a modern mobile convolutional object detector.
- We show that this relationship stays similar after transfer learning a different dataset.
- Our results show that the precision/object size curve might give extra information about the performance of a network compared to just the AP over a benchmark dataset.

2. Related Work

Comparing the performance of different networks can be difficult. This fact is for example the case for inference speed: papers for techniques like Pelee [8] and YOLO [5] do tend to publish that they achieve a certain frame rate, and compare this framerate to other networks. However, these measurements are usually conducted under different conditions, such as different feature extracters, hardware, input image sizes and software platforms. Because of these varying conditions, plotting speed/accuracy tradeoffs can provide more information about an architectures performance than an one-dimensional framerate measurement. Measuring these tradeoffs under standarized conditions allows a fair comparison between networks for speed, as was done by Huang et al. [9].

For precision, large public datasets and detection challenges like COCO [6] and PascalVOC [10] allow developers to make a comparison on AP between the different networks. Most papers report their achieved precision on one of these datasets. This precision is calculated as

$$\text{Precision} = \frac{\text{True Positives}}{\text{All Positives}}. \quad (1)$$

The precision of detections is usually reported for a certain Intersection over Union (IoU), which describes how much the detected bounding box overlaps the ground truth bounding box, as calculated by:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \quad (2)$$

For the COCO challenge, the primary challenge metric is the AP averaged across 10 different IoU thresholds between .5 and .95, with a step size of .05 [11].

Some papers investigate the relationship between object size and precision. The effects of the bounding box area on the precision per category of the PascalVOC dataset for five size groups (XS, S, M, L and XL) is investigated in the SSD paper [4]. As is shown in figure 2, this research shows that object detectors perform better on larger objects.

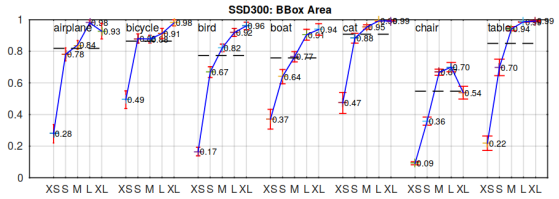

Figure 2. The effects of bounding box area per category for SSD300 [4]

As is shown in figure 3, bounding box area might however not be related to the actual object pixel area and could therefore not give a precise picture of the object size versus precision trade off.

Figure 3. A giraffe and its bounding box. The actual object only covers a small area of the bounding box. Adapted from [12]

As is shown in figure 4, object segmentation area represents the actual amount of pixels that make up an object in an image better than bounding boxes.

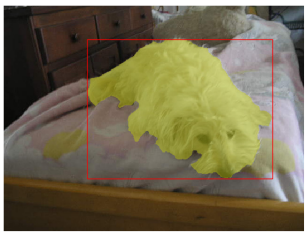

Figure 4. A dog and its boundingbox and segmentation from the Pascal In Detail dataset [13].

The COCO dataset is fully segmented and its evaluation API calculates precision for three segmentation area scales by default[2]. Not all papers that measure their precision on COCO state their results for the different object segmentation areas. For instance, in the paper for the network we use, no performance is given for the different object areas. Some papers, such as YOLO9000 [14] and R-FCN [15], do compare their results to others for these different scales. A comparison between the accuracy on large and small objects for different networks is made by by Huang et al. [9].

[1] MobileNetV2 + SSDLite [7]

[2] small (area $< 32^2$), medium ($32^2 <$ area $< 96^2$) and large (area $> 96^2$), in pixels

However, using only a few datapoints to evaluate the object area/precision relation of a network might be enough resolution to show the network performance, as we show in this paper.

3. Methods

We measure the precision of a real-time object detector for mobile platforms, MobilenetV2+SSDLite [7]. In our experiments we use a network that is pretrained on the COCO dataset.[3] This network has an input image size of 300 by 300 pixels.

For inference, we use the Tensorflow object detection API [9]. We infer detections on the COCO val2017 [6] dataset of this architecture trained on the COCO dataset.

To measure the relationship between object size and precision, we use the COCO API [16]. Instead of the default small, medium and large areas scales, we divide all objects over 100 area buckets, with each bucket containing an equal amount of objects. We then calculate the average area of a bucket, and calculate the average precision for every bucket.

We create our own dataset of the logo of Aiir Innovations. For this we have taken 80 clips in different settings, distances and lighting conditions. We have annotated these videos by hand using a video masking tool. From these clips, we extract one frame for every second of video, resulting in 542 images. The gamma of the images is changed randomly, in order to create more variation in the dataset. In figure 5 we show an example of our created dataset.

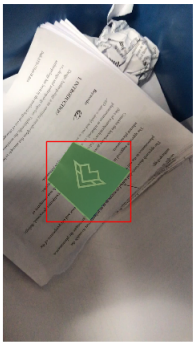

Figure 5. An example of an image from our own dataset of the logo of Aiir Innovations, showing the bounding box and segmentation annotation (note that the Aiir logo is blue and color changes are due to the displayed segmentations).

We add this extra class, called "Aiir", to the Pascal in Detail [13] dataset, which contains fully segmented objects for the original PascalVOC2010 dataset. We then convert these annotations to the COCO annotation format. Finally, we train the MobilenetV2+SSDLite network on this new dataset using transfer learning for 191 epochs on a batch size of 50, and measure the precision for the validation set using the COCO API.

[3] As taken from `http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz`

4. Results

In order to rule out that our network performs less well on small objects because it was only trained on large ones, we check that our training datasets contain more small objects than large ones. As is shown in Figure 6 and Figure 7, both training sets contain many more small object than large objects. Overall, the objects in Pascal in Detail + Aiir train are larger than those in COCO train2017.

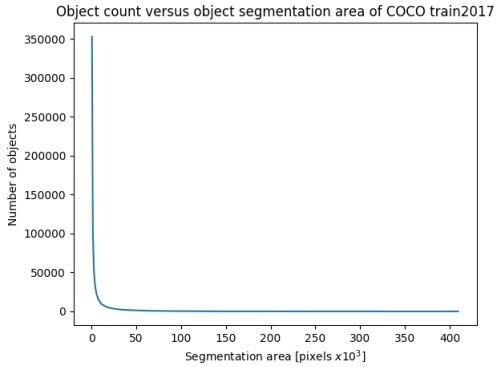

Figure 6. The object size distribution of the COCO 2017 train set.

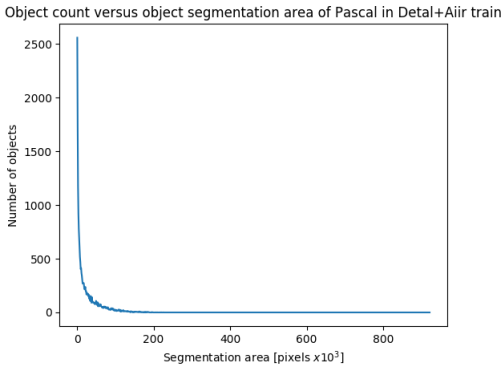

Figure 7. The object size distribution of our Pascal in Detail + Aiir train set.

As shown in table 1, the COCO training dataset is much larger than Pascal in Detail+Aiir. Furthermore, it should be noted that due to an error the generating of our dataset, the Pascal in Detail + Aiir train set misses 20% its object annotations, uniformly distributed over its classes and object sizes. This lowers our overall achieved AP, but since it is uniformly distributed over object sizes, it should not influence the object size/precision relation.

Table 1. Image splits for the used train and validation sets

	# images train set	# images val set
COCO 2017	118k	5k
Pascal in Detail + Aiir	9k	1,6k

4.1. Object size versus precision for COCO val2017

The network should perform better on large objects than small objects. As Figure 8 shows, the average precision of the network rapidly increases until an area of around $50{*}10^3$ pixels. The upper bound of the AP is around 0.7 for this network and dataset. Note that the largest image size in the coco dataset is $410 * 10^3$ pixels.

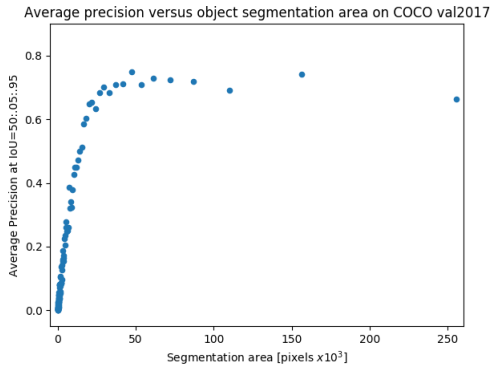

Figure 8. AP at IoU = [0.5:0.05:0.95] versus object segmentation size for the COCO validation set. The achieved AP over the entire dataset is 0.243.

As is shown in figure 8, there is a strong relationship between object size and precision that is not shown in the singular value of AP over the entire dataset.

4.2. AP over entire COCO val2017 dataset

The COCO val2017 dataset has a similar object size distribution as its training counterpart, as is shown in Figure 9. Therefore most objects in the COCO val2017 set are of a size that our network performs less well on, and the AP over the entire COCO validation set should be much lower than the networks accuracy upper bound of 0.7. Using the COCO API we measure an overall AP of 0.243.

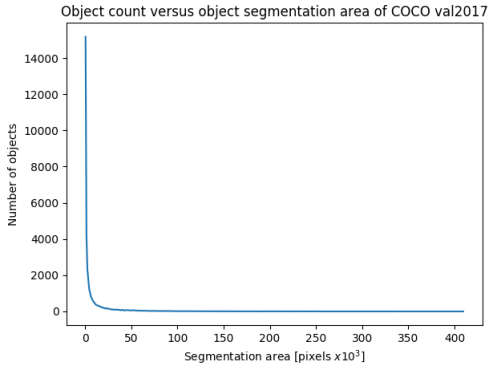

Figure 9. The object size distribution of the COCO 2017 val set.

4.3. Object size versus precision for Pascal in Detail+Aiir val

To verify that the same network shows a similar object size/precision relationship for a different dataset, we first train our network on the Pascal+Aiir classes using transfer learning. We then measure the performance of the network on our validation set for Pascal in Detail+Aiir.

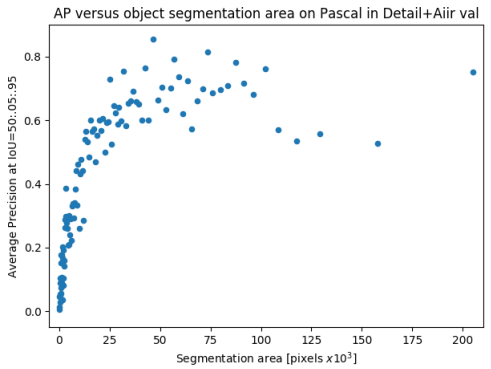

Figure 10. AP at IoU = [0.5:0.05:0.95] versus object segmentation size for the Pascal in Detal+Aiir validation set. The achieved AP over the entire dataset is 0.403.

As Figure 10 shows, the network shows a strong object size/precision relationship for this dataset as well. The data points show more variance than those for coco, which could be caused by the validation set being smaller.

4.4. AP over entire Pascal in Detail+Aiir val dataset

Because the object size and precision tradeoff for the Pascal in Detail+Aiir is similar to that for COCO, the object size distribution of the inferred dataset should give an indication of the AP over the entire dataset.

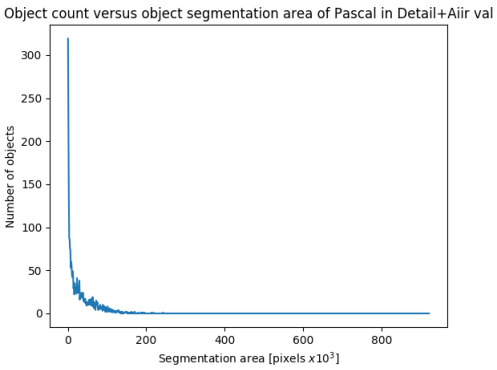

Figure 11. The object size distribution of the Pascal in Detal+Aiir val set.

Figure 11 shows that the Pascal in Detail+Aiir validation set contains a larger objects than COCO val2017. Therefore, the measured AP over the entire dataset should be higher than the AP measured over COCO. Using the COCO API, we determine that the COCO primary challenge metric achieved by the network is 0.403, confirming this theory.

5. Discussion

Our results show that the precision of the MobilenetV2+SSDLite architecture is dependent on the size of the objects it is asked to infer. Because of this, the AP over an entire dataset for a given network is highly dependent on the object size distribution of the dataset used.

The relationship between object size and precision seems to be similar for different datasets when using transfer learning. However, this might not hold when training networks from scratch. The results of our transfer learned Pascal in Detail+Aiir dataset also show more variance than those for COCO. This could be because the network is not optimally trained, as 20% of object annotations in the training set are missing.

The relationship between object size and precision might be influenced by various measurement conditions, such as input image size and network design. This curve might not only have different *values* for different conditions, but also different shapes. Performing the same type of experiments on multiple architectures and databases in standarized conditions, much like [9], could provide more information on the merits of evaluating the performance of networks using object size/precision curves.

6. Conclusion

We have shown that the precision of a convolutional object detector is highly dependent on object size. We have also shown that because of this the object size distribution of a dataset influences the overall AP of a network for a given dataset. Combining these results, we can conclude that the object size/precision curve of a network for a dataset might give more detailed information about the performance of an object detector than AP over an entire dataset.

References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[2] Ross Girshick. Fast r-cnn. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 1440–1448. IEEE, 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[8] Robert J. Wang, Xiang Li, Shuang Ao, and Charles X. Ling. Pelee: A real-time object detection system on mobile devices, 2018.

[9] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7311, 2017.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[11] Coco detection metrics. http://cocodataset.org/#detection-eval.

[12] John Hilliard. Giraffe, 2010.

[13] Cvpr'17 pascal in detail challenge. https://sites.google.com/view/pasd. Accessed: 2018-06-23.

[14] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. IEEE, 2017.

[15] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 379–387. Curran Associates, Inc., 2016.

[16] Coco api. https://github.com/cocodataset/cocoapi.