OBJECT DETECTION BASED ON CONVOLUTIONAL NEURAL NETWORKS TRAINED ON SYNTHETICALLY GENERATED DATA

Depascuali, Francisco

Instituto Tecnológico de Buenos Aires (ITBA), Buenos Aires, Argentina
Fachhochschule Technikum Wien, Vienna, Austria

Abstract

Convolutional Neural Networks (CNNs) have been consolidated as a powerful statistical tool, achieving the best benchmarks in the recognized computer vision contests PASCAL VOC, COCO, and ILSVRC. It has been shown that when there is enough training data available, even simple models can address image classification problems with outstanding results. However, the required data for real-world scenarios tends to be unavailable or hard to obtain in large quantities. This paper address the problem of insufficient data by exploring synthetic data generation tools for training purpose, and analyze how accurate does this training results in real world object detection.

Introduction

One of the main goals of computer vision algorithms is the task of classification: given an input image, output the corresponding class. Former approaches to classification were based in detection of engineered features, such as SIFT[1] and HOG[2].

It was on the year 2012 that a type of statistical models, Convolutional Neural Networks (CNNs), had its breakthrough winning the 2012 ILSVRC contest with an error rate of 15.4%. The model proposed, AlexNet, was consolidated as the foundation of modern research in CNNs [3].

Throughout the years, new variations of CNNs were designed, which lead to RESNET achieving an error rate of 3.6% in ILSVRC 2015 (human vision obtain between 5% and 10% error rate) [4].

With those outstanding results in classification, the focus shifted into object detection, the task of classifying and localizing multiple classes in a given image.

Region-based CNNs

The main family of CNNs architectures used in object detection is called region-based CNNs. The idea behind this type of CNNs is to propose different regions in the image and classify each of them for the presence of an object of interest.

This type of algorithms focus on the step of localization, while using already known architectures (such as VGG[5] or RESNET[4]) for the classification phase.

The metrics used is accuracy and performance. Performance is measured in FPS (frames per second), while Accuracy is measured in mean average precision (mAP [6]). The main examples of this family of algorithms are R-CNN (and its variants), SSD and YOLO.

Synthetic Image Generation

For CNNs to work, large amount of annotated data is required. However, availability of annotated datasets is low.

Different approaches have already been used for synthetic images generation. A common technique is data augmentation**,** which consists in generating a new dataset by modifying the training images. These modifications usually include horizontal flipping, random crops, color jittering, rotating, mirroring and adjusting contrast.

A widely used technique is fancy PCA, proposed by Krizhevsky et al. [3] for training AlexNet, which alters the intensities of the RGB channels during training. More advanced approaches, such as [7], create images by identifying objects of interest and relocating them onto different background.

The main goal is to make synthetic images as real as possible in the semantic perspective, that is, that can represent the features of real images, taking into account different aspects (such as colors, occlusion, pose, background or shadows) of

a real world scenario. Consequently, arbitrarily sized datasets could be generated for both training and testing.

Materials and methods

Object detection algorithm

This main alternatives analyzed were YOLO [9] and SSD [8], because of their results in VOC [6] datasets. Though the accuracy in VOC 2007 is similar for both methods (see Table 1), a qualitative analysis for car detection was done with both of them. SSD showed better results regarding the stability of the detected bounding box.

Another important point is that SSD is implemented on top of a well known framework for deep learning, caffe. It already implements tools regarding performance, debugging and network visualizations.

The downside of this choice, however, is that YOLO v2 shows greater performance.

As accuracy and stability of bounding box is critical for the application of this paper, SSD was chosen.

It is important to remark that the main focus of this paper is on synthetic dataset generation, which means that the techniques described can be applied to both YOLO and SSD, and other types of object detection algorithms.

Algorithm	Input size	mAP	FPS
SSD 512	512 x 512	79.8	19
SSD 300	300 x 300	77.2	46
YOLO v2	448 x 448	76.8	67
YOLO v2 544	544 x 544	78.6	40

Table 1: Comparison of SSD and YOLO trained and tested with VOC 2007.

Synthetic Image Generation tool

One valid and promising approach for cars would be to create a scene with a city (or play a driving game) and drive the car around while modifying the camera position. Whenever needed, a new image can be rendered. This idea was explored, but no high-quality cities (or game engines) were found.

The approach chosen is to create synthetic images similar to [7] and [10], with the object and the background in a 3D world. To be able to create the 3D world, an open-source tool called blender is used, with a scene that contains both the 3D object and the background. The tool used (which is already implemented by [10]) generates images with a ground and a sky.

There is one main point that differs between object detection and viewpoint estimation [10]: the influence of the background is greater in object detection. This makes the task of object detection harder, and therefore bigger datasets are needed to counter the challenges of different background conditions.

Fig. 1: Object detection and viewpoint estimation input images. Note the presence of background in left image.

Ground and sky

Ground and sky is a scene modeled by a plane and a background, and their texture can be switched by scripting. It is important to mention that there should be a considerable amount of textures, and different between training and testing. Combined with varying light conditions, this will help the model to avoid overfitting on already known textures.

Fig. 2: Ground and sky example images.

Whole background

A scene with a single background image and the model was implemented for this paper. The reasoning behind this kind of images is that region-based CNNs sample regions from the image, taking negative examples from the background (which aren't labeled as part of the 3D model). Therefore, there is more real information by using whole real images as background, taking into account that the position of the car in the image isn't relevant for the algorithm.

To obtain the background images, a tool for downloading batches of 100 images from google search was developed. In this paper, 10 different categories were used (1000 different backgrounds) relative to the 3D car, for example street and city.

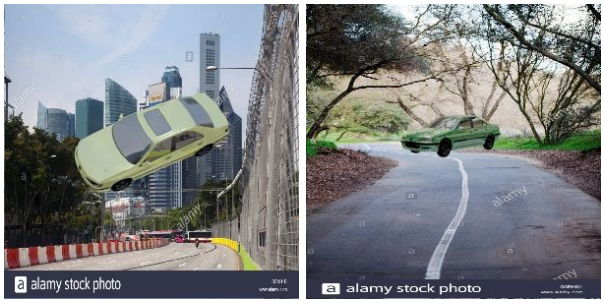

Fig. 3: Whole background example images.

Quantitative Analysis

As already mentioned in the problem definition, representative images for real datasets are insufficient or unavailable. This makes it difficult to quantitatively test the train generated datasets.

Therefore, using the same strategy as [10], a synthetic dataset is also generated for testing. This test dataset is of the same type as the train dataset (whole background or ground and sky) but different textures, light conditions and angles are used for generating it.

Train dataset size	Test accuracy
50k	0.96 ± 0.007
100k	0.98 ± 0.001
150k	0.98 ± 0.005
200k	0.99 ± 0.004

Table 2: Training with ground and sky background for 50k iterations.

Train dataset size	Test accuracy
50k	0.90 ± 0.02
100k	0.91 ± 0.03
150k	0.93 ± 0.01
200k	0.94 ± 0.02

Table 3: Training with whole background for 50k iterations.

Qualitative Analysis

A more interesting aspect is the qualitative analysis, that gives more insight in how the generated dataset helps the network to detect cars in real world scenarios.

Ground and sky

The model trained with 200k images of ground and sky was used to evaluate car detection in real world images.

The detections didn't correspond to the accuracy shown in the quantitative analysis.

Fig. 4: Incorrect detections using ground and sky model.

Whole background

Images and video sequences were analyzed using the whole background model trained with 200k images, to evaluate car detection in real world scenarios.

The detector achieves the best accuracy with a golf model with white uniform background and a good image quality.

Fig. 5: Correct classification of vw golf. For the image in the right, the bounding box is not completely correct.

Fig. 6: Incorrect classification of vw golf. In left image, a wheel is detected as a car. To the right, a peugeot taxi is classified as a golf.

Discussion

Ground and sky

The model learned to classify synthetic images, but failed for detecting real ones.

After analyzing this results, it can be seen that the network overfitted in the training dataset. New experiments should be created by training for less iterations.

Whole background

In both images of Fig. 5, the classification is correct, while in the right image the bounding box is misplaced. The hypothesis is that it is related to the dataset generation tool, because it generates images from different perspectives with a uniform probability distribution. This means that there are similar number of images from different distances and perspectives, while the network needed more close-up images of the car for detection in close ups.

Regarding incorrect classification (Fig. 6), the wheel classified as a car was an unexpected result. This could be related to all images having wheels, therefore producing a high activation map on the convolutional layers. A possible solution would be to train the detector on the wheel class as a separate one, and also to occlude the wheels in some of the training images.

For the image at the right of (Fig. 6), it is important to mention that distinguishing between different types of car is a difficult problem, and with just training on one class, it is highly probable that when shown other cars, the model predict that it is a golf.

It is relevant to note that all the regions in the images were there isn't a bounding box, describe the effectivity of the model in avoiding to detect cars where there aren't (false positives).

As shown in Fig. 5, the detector achieves a good performance in the absence of a background.

Conclusion

This work describes an approach to train an object detector with synthetically generated images. This detector prove to have accurate results detecting a single class, but it shows constraints related to the background.

For more robustness, other 3D models should be added, which will help the network to distinguish between those different types, avoiding false positives.

It is really promising that the network achieved this results trained with synthetically generated images from scratch. Here arise the most interesting aspects. One of them, is that more synthetic images could be generated, and will make the object detector better. On the other hand, instead of training the network from scratch, an already trained network (with real images) in car detection could be used, fine-tuning it with synthetically generated datasets of the desired model. This way, the general characteristics would already be assimilated by the network (transfer learning), making it possible to learn the details of the model.

References

[1] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comp., 1989.

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

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

[4] K. He, X. Zhang, S. Ren and J. Sun. Deep Residual Learning for Image Recognition. arXiv preprint, arXiv:1512.03385, 2015.

[5] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint, arXiv:1409.1556, 2014.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV, 2010.

[7] Rozantsev, Artem; Lepetit, Vincent; Fua, Pascal. On Rendering Synthetic Images for Training an Object Detector. eprint arXiv:1411.7911, 2014.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: Single shot multibox detector. arXiv: 1512.02325, 2015.

[9] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR, 2016.

[10] Tylson, Emilio. Domain-Specific Object Viewpoint Estimation based on Convolutional Neural Networks Trained on Synthetically Generated Data. 2017.

Author's address

Francisco Depascuali
se16m501@technikum-wien.at