Master's Thesis

Authors:
Christoffer Bøgelund Rasmussen

Supervisors:
Kamal Nasrollahi

Aalborg University
VGIS
10th semester
Group 17gr1041

TBA

AALBORG UNIVERSITY

Title:
Master's Thesis
Subject:
Vision, Graphics and Interactive Systems
Project period:
1/2-2017 to TBA
Project group:
17gr1041
Participants:
Christoffer Bøgelund Rasmussen

TBA

Supervisor:
Kamal Nasrollahi
Printed copies:
TBA
Number of pages:
TBA
Appendix media:
AAU digital exam zip file
Finished:
TBA

Preface

Reading Guide

Tables, code listings and figures are numbered sequentially within each chapter. Citations are written as [x] where x denotes the reference number used in the bibliography. Code classes and functions are written as `class` and `function()`, respectively. Additional files have been uploaded to the AAU Digital Exam.

Christoffer Bøgelund Rasmussen

Contents

1 Introduction

Object detection is a fundamental area of computer vision that has had a great amount of research over the past decades. The general goal of object detection is to find a specific object in an image. The specific object is typically from within a pre-defined list of categories that are of interest for a given use case. Object detection generally consists of two larger tasks; localisation and classification. It is assumed that the objects of interest are not already located in the image and as objects can vary in number of pixels depending on factors such as distance and scale, objects must be both localised in an image and classified accurately. Localisation is typically done by with a bounding-box indicating where a given object is in the image. However, other methods such as objects' centres and closed boundaries can also be used. Not only is object detection an important task in localising and classifying, it is also a necessary earlier step in larger computer vision pipelines. For example, object detection is needed within the tasks such as activity and event recognition, scene understanding, and robotic picking.

Object detection is a challenging problem due to both some large scale issues and minute differences. Firstly, there is the challenge of differentiating objects between classes. Depending on the problem at hand the sheer number of potential categories present can be into the thousands or tens of thousand. On top of this separate object categories can be both very different in appearance, for example an apple and an aeroplane, but separate categories can also be similar in appearance, such as dogs and wolves.

Current state-of-the-art within object detection is also within the realm of deep learning with Convolutional Neural Networks (CNNs). This is exemplified with almost all entries in benchmark challenges such as Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes (PASCAL VOC) [1], ImageNet [2], and Microsoft Common Objects in Context (MS COCO) [3] consisting of CNN based approaches. However, improvements are still needed before object detection can be used in real-world scenarios that require a high level of precision, accuracy, and performance.

1.1 Initial Problem Statement

An initial problem statement can be formed as follows:

> *What are the specific challenges within object detection?*

Based upon this, Chapter *2 Problem Analysis* will outline these challenges. On top of this, related work into object detection, both current state-of-the-art and also classic methods, will be researched.

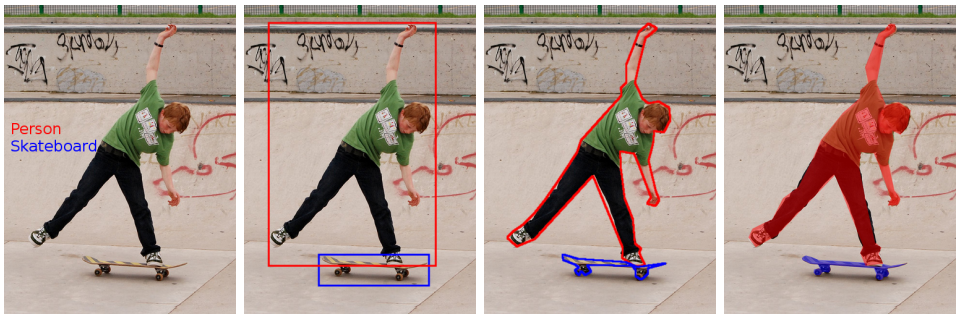
2 Problem Analysis

This chapter will outline object detection and it's key challenges. This includes aspects within robustness, computational-complexity and scalability. Once completed the key works within object detection will be analysed, both current state-of-the-art and notable older methods.

2.1 Object Detection

As mentioned in Section *1.1 Initial Problem Statement*, object detection consists of two larger tasks; classification and localisation. Depending on the problem at hand, object detection can be split into two categories. If only a single class is of interest, such as detecting a specific traffic sign, the object detection task is denoted as class-specific detection. Whereas, the more general case when multiple classes are of interest in an image is denoted as multi-class detection [4]. Key challenges such as PASCAL VOC, ImageNet, and MS COCO are of the latter task. This thesis will be within the multi-class detection domain and take these challenges into account when analysing related works in Section **?? ??** and determining the algorithm to be implemented and evaluated in Section **?? ??**. An analysis of these key challenges is done in Section **?? ??**. The goal of a detector is to output a list of labels from a predefined list of categories indicating which objects are present and where they are located in an image. Object detection has a number of related fields which share to common goal of categories relevant objects. This can be seen in Figure 2.1. In all four instances the goal is to categorise the two objects person and skateboard, however, the difference lies in the level of localisation precision. In Figure 2.1a, object categorisation aims to only classify the objects in the image without providing any indication as to where the objects are located. Object class detection in Figure 2.1b, localises the classified objects with the use of bounding-boxes, where ideally the bounding-boxes are placed as tightly around the given object as possible. Figure-ground segmentation in Figure 2.1c, indicates localisation with a lasso outline around the objects. Finally, in Figure 2.1d, semantic-segmentation localises objects at a pixel-level classifying each pixel that is related to the given object.

correct section refs to above



(a) **(b)** **(c)** **(d)**

Figure 2.1: Example of vision tasks related to object detection. All tasks have the common goal of categorising predefined objects. Methods are: object categorisation (a), object class detection (b), figure-ground segmentation (c), semantic Segmentation (d). Image and class labels taken from MS COCO [3].

A more recent example of segmentation is that of instance segmentation. Instance

segmentation varies to semantic segmentation in that individual instances of objects are classified as such. If multiple instances of the same object is present, such as an image of a crowd with many people, in semantic segmentation all people will be given the same label as one large group. However, in instance segmentation the people are still given the same label but individual instances of a person is also found. This area of research within segmentation is relatively new, however, is beginning to become more popular in comparison to semantic segmentation. For example, the MS COCO segmentation challenge which has been held in 2015 and 2016 only accepts instance segmentation entries.

2.2 Main Challenges

The challenges of object detection can be split into two groups as per [4]:

- Robustness-related.
- Computational-complexity and scalability-related.

The following sections will outline the above.

2.2.1 Robustness-related Challenges

Robustness-related refers to the challenges in appearance within the both of intra-class and inter-class. Intra-class is the differences in appearance of objects which are of the same class. For example as seen in Figure 2.2, all of the images belong to the superclass chair from the ImageNet training set [2], however, vary greatly in their overall appearance.

(a) **(b)** **(c)** **(d)**

Figure 2.2: Examples of intra-class appearance variation. All images have the label chair in the ImageNet training set [2].

An object detection system must be able to learn

explanation that typically obj detection is supervised learning

the appearance variations that can occur intra-class. These variations can be categorised into two types as per [5]:

- Object variations.
- Image variations.

Object variations consist of appearance differences between instances of colour, texture, shape, and size. Image variations are differences not related to the object instances themselves but rather consist of conditions such as lighting, viewpoint, scale, occlusion, and clutter. Based upon these conditions the task of both classifying a given object as a given class but also differentiating the potentially largely varying objects into the same class challenging.

unsure if to add explanation of structured and unstructured classes

Robustness-related challenges can also occur with inter-class appearance differences. This refers to the differences between objects that are regarded as different categories. Challenges arise in scenarios where an object detector must decide if an instance is between classes that are very similar. For example using images and their respective classes from ImageNet [2], in Figure 2.3 and Figure 2.4 the differences between the two examples are very similar, however, their class labels are different. In Figure 2.3a and Figure 2.3b the class labels are mini-bus and delivery truck respectively. In Figure 2.4a and Figure 2.4b the labels are white wolf and German shepherd.

(a) **(b)**

Figure 2.3: Examples of inter-class appearance variation. Both images are from the ImageNet training set [2] and have the labels mini-bus (a) and delivery truck (b).

(a) **(b)**

Figure 2.4: Examples of inter-class appearance variation. Both images are from the ImageNet training set [2] and have the labels White wolf (a) and German shepherd (b).

It should be noted that this is a task-specific if inter-class appearance similarities is a problem or not. It can be argued that both the examples in Figure 2.3 and Figure 2.4 can be grouped into a larger superclass label if the given task does not require training of a model to such granularity. In both examples the classes stated are of the lowest class available in the overall hierarchy. ImageNet has labels available for each image along a larger array of classes and sub-classes. Figure 2.5 visualises the granularity possible where both images belong to the superclass animal.

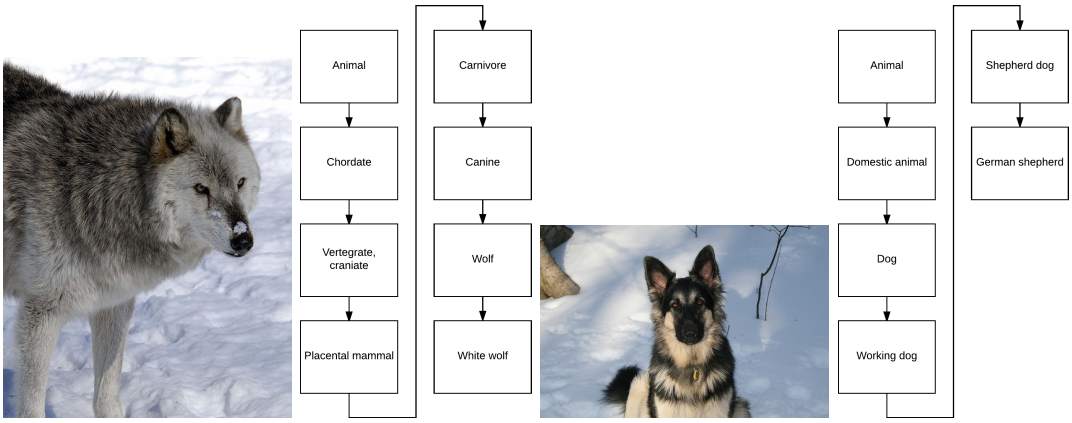


Figure 2.5: Visualisation of the hierarchy of potential classes for two examples in the ImageNet training set [2].

2.2.2 Computational-complexity and Scalability-related Challenges

The second challenge as per [4] is related to the potential scale of object detection. When deciding on which type of model to use it must be complex enough to be able to capture the previously mentioned challenges both in inter- and intra-class. On top of this there is an extremely large number of potential classes in object detection. If the aim is to train

a model to classify between an extreme number of classes then naturally a large number of images are also needed for each category. The large number of images need also to be representative enough in training to capture the necessary visual features to generalise on non-training images. In 2016 the ImageNet object detection challenge there is a total of 200 object categories, with 456,567 images comprising the training set. Therefore, a complex enough model is needed in order to learn and generalise on such a dataset but this of course places high requirements on the amount of training needed.

Issues can also arise over time when designing an object detection system. Over time the visual appearance of an object can change, which is very difficult to take into account when training a model. For example, the visual appearance of televisions have changed greatly in the past century. If a system were only to be trained on images from an earlier time period it may not be able to generalise on new instances. Therefore, it is important that a model is able to be updated as the appearance of objects change. On top of this, new categories of objects can come to fruition which may needed to be added to a model.

2.3 Implementation Outline

As per [4] the steps in the general pipeline for an object detection system is as follows:

1. Find all possible object regions in the image.
2. Determine if the regions correspond to any of the predefined categories.
3. Evaluate all responses from step 2 to determine final detections.

2.4 Related Work

One of the first methods to show that CNNs could significantly improve object detection was that of R-CNN [6]. The method obtains the name R-CNN based upon a CNN is used on regions of the image. Many earlier object detection approaches were used in a sliding window fashion testing all areas of an image. This can lead to a huge amount of potential testing windows especially if the object detection is done at a multitude of different scales. The method was heavily inspired by the AlexNet model that started the deep learning renaissance in 2012 by winning classification challenge in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The authors of R-CNN aimed to show that the advances in classification with a model such as AlexNet could also be done in object detection. In R-CNN the model is used as a feature extractor from which a class-specific linear Support Vector Machine (SVM) can be trained on top of. The AlexNet-based feature extractor is firstly pre-trained on a large dataset designed for classification, in this case the training set from ILSVRC 2012. This pre-trained model is then adapted to the new domain of object detection by fine-tuning the model accordingly. In this instance the authors fine-tuned warped training instances from the PASCAL VOC dataset. The AlexNet model was also altered to classify the 20 classes present in PASCAL VOC rather than the 1000 classes in ILSVRC. The pipeline of the R-CNN is split into 3 modules:

1. Region proposals.

2. Feature extraction.

3. Class-specific linear SVMs.

In this first module, region proposal, there is a large number of choices of methods to produce a suitable number of windows in comparison to a sliding window approach. R-CNN is agnostic to the region proposal method chosen and in the original work SelectiveSearch [7] is used. Module two, as explained earlier, is the use of a CNN as a feature extractor. This is in the form of a 4096-dimensional feature vector from the domain-specific PASCAL VOC trained AlexNet model. These feature vectors are used in the third module, class-specific linear SVMs. In the case of PASCAL VOC a total of 21 SVMs are trained, one for each of the 20 classes in the challenge and one for a background class. The training of the SVMs is done by forward propagating a large number of both positive and negative region proposals found with SelectiveSearch and storing each 4096-dimensional feature vector to disk. After this the appropriate labels are applied to each vector and a linear SVM is optimised for the 21 classes. At test time, for a given image SelectiveSearch is used to produce around 2000 proposals. Each of the proposals are propagated through the network to extract their respective feature vectors. Each feature vector is then tested against every SVM to produce a score for each class. Finally greedy Non-maximum Suppression (NMS) is applied to remove overlapping detections. The approach outlined in R-CNN produced a significant improvement in object detection with an improvement of roughly 13%, compared to previous state-of-the-art methods, to an overall 53.7% Mean Average Precision (mAP) on the PASCAL VOC 2010 test set. Similar results were also found on the PASCAL VOC 2011/12 set with mAP of 53.5%. Despite the significant improvements with a CNN-based method on region proposals there are still issues with the R-CNN. Firstly, the testing time per image is very slow at roughly 47 seconds on an Nvidia K40 GPU. Also extracting features for each proposal in order to train the SVMs takes a large amount of disk space and may not be feasible on all hardware configurations. Finally, as the R-CNN is made up of 3 modules the training is done in a multi-stage manner rather than end-to-end. Therefore, the loss calculating when optimising the SVMs are not used to update the CNN parameters.

The R-CNN method was improved the following year with Fast R-CNN [8] and aimed to improve speed and accuracy. One of the significant changes is that the detection training done is now end-end rather than in the multi-stage pipeline in R-CNN. Due to this the large requirements of disk space due to feature caching is no longer required. The Fast-RCNN method takes both an image and a set of pre-computed object proposals. A CNN forward propagates the entire image, rather than individual proposals in R-CNN, through several convolutional and max-pooling layers to produce a feature map. Features are extracted for each proposal in their corresponding location in the computed feature map with a Region of Interest (RoI) pooling layer. The RoI feature is calculated by splitting the $h \times w$ proposal into $H \times W$ sub-windows of size $h/H \times w/W$. Where h and w is the height and width respectively of a proposal. H and W are hyper-parameters specifying the fixed spatial extent of the extracted feature. Each sub-window has max-pooling applied and with the resulting value being placed in the corresponding output cell. Once the RoI pooling layer has been applied to a pre-computed object proposal the forward pass continues through two fully-connected layers followed by two sibling output layers. The sibling outputs are a softmax classification layer that produces probabilities for the object classes and another

Table 2.1: A comparison of R-CNN and Fast R-CNN PASCAL VOC mAP results on the test set from 2007, 2010, and 2012.

	2007	2010	2012
R-CNN	66.0	62.9	62.4
Fast R-CNN	**66.9**	**66.1**	**65.7**

Table 2.2: Speedup between R-CNN and Fast R-CNN in regards to both training and testing. Both methods are train a VGG16 network for object detection.

	R-CNN	Fast R-CNN
Train time (hours)	84	**9.5**
Train speed-up	1x	**8.8**×
Test time (seconds/image)	47.0	**0.32**
Test speed-up	1x	**146**×

layer for bounding-box regression. These two layers replace the respective external modules in R-CNN and make it possible to train the entire detection network in a single-stage. As in R-CNN, pre-training a CNN on a large classification dataset and fine-tuning towards detection and a specific object classes is done in a similar fashion. In R-CNN, the only deep network used was AlexNet [9], however, in Fast R-CNN the authors experiment with networks of different size. It was found that the deeper network VGG16 [10] for computing the convolutional feature map gave a considerable improvement in mAP. For a fair comparison of results against R-CNN, its CNN was the same pre-trained VGG16 network. The mAP results on PASCAL VOC 2007, 2010, and 2012 compared against R-CNN can be seen in Table 2.1.

However, as the name Fast R-CNN implies the main improvement is the speed in respect to both training and testing. By computing a convolutional feature map for an entire image rather than per object proposal the number of passes in the network is lowered significantly. Also by makign the detection training end-to-end with the two sibling layers lowers the training time needed. An overview of time spent training and testing both Fast R-CNN and R-CNN with a VGG16 CNN can be seen in Table 2.2.

While Fast R-CNN provided improvements in both accuracy and speed, the increase in speed is only in relation to the actual object detection and assumes that the region proposals are pre-computed. Therefore, there is still a significant bottleneck per image as a region proposal method can typically take a couple of seconds.

The region proposal bottleneck was addressed in the third iteration of the R-CNN network with Faster R-CNN [**?**]. In this method it was shown how to compute region proposals with a deep CNN with a part of the network called Region Proposal Networks (RPN). A RPN shares the convolutional layers and feature map used for computing features with RoI pooling in Fast R-CNN. As this deep network is already being computed on the entire image for the classification pipeline the added time for proposals using the RPN is negligible (10ms) in comparison to a method such as SelectiveSearch. Apart from the change in how region proposals are computed there is no change in comparison to Fast R-CNN. An RPN takes the convolutional feature map as input and returns a number of object proposals. Each proposals is fed into two sibling layers, similar to that in Fast R-CNN, one layer scoring how

likely to be an object or background and another performing bounding-box regression. The proposals are found through a method denoted as anchors. At each sliding-window location k proposals are with anchors that are user-defined reference boxes for how an object proposal may be formed. The anchors can be built based upon scale and aspect ratio altering the size. In the original work three different scales and three aspect ratios are built, yielding a total of $k = 9$ anchors at each sliding window position. These anchors are then placed on the feature map and the sibling layers calculate the likelihood of an object and regress the anchor as necessary. Once the proposals have been found with the RPN these are placed on the same convolutional feature map as earlier and the rest of the pipeline is identical to Fast R-CNN, classifying and regressing bounding-boxes with another set of sibling layers. As the only change is the addition of computing proposals in the network with RPN, the results are similar in respect to mAP. Only a slight improvement in made on PASCAL VOC 2007 and 2012, from 66.9% to 69.9% and 65.7% to 67.0% respectively. However, the main contribution to the work is the speed-up of the entire object detection pipeline as the object proposal time is now minimal. On average processing an image on PASCAL VOC 2007 with an Nvidia K40 with Fast R-CNN including proposals took 2 seconds per image. While in Faster R-CNN with the same hardware takes 0.2 seconds per image. A speed-up of 10× from Fast R-CNN to Faster R-CNN and 250× from the original R-CNN. The Faster R-CNN methods has also proved to be the foundation for the winning entry in multiple detection challenges including MS COCO. The results for this challenge with a VGG-16 model for Fast R-CNN were 35.9% mAP@0.5 Intersection-Over-Union (IoU) and 19.7% mAP@[0.5, 0.95]. Faster R-CNN improved this to 42.7% mAP@0.5 and 21.9% mAP@[0.5, 0.95].

3 Technical Analysis

4 Discussion

5 Conclusion

Bibliography

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10602-1_48

[4] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 10:1–10:53, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2522968.2522978

[5] F. Schroff, *Semantic Image Segmentation and Web-supervised Visual Learning*. University of Oxford, 2009. [Online]. Available: https://books.google.dk/books?id=4EqZYgEACAAJ

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11263-013-0620-5

[8] R. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

Appendices

Notes